
ATARI
ST
COMPUTER
Die Fachzeitschrift für den ATARI-ST Anwender.
September '87
9
DM 6,–
Ös 53,–/Sfr. 6,–
NUR AUFGEBLASEN?
Der neue Mega ST
K-MAX:
Erste Transputerkarte
für den ST
APL 68000:
Eine Zeichensprache
OMIKRON BASIC
COMPILER!
NEU+++NEU+++NEU+++NEU+++NEU+++NE
GRAFIK KURS
1. TEIL

Das Bildungswesen und der Computer

Die Geschichte der Neuzeit ist geprägt vom zunehmenden Einfluß der Computer auf viele Bereiche unseres Lebens. Was sich einst nur auf die Produktion von Industriegütern beschränkte, erobert immer größere Anteile unserer privaten Sphäre. Ein Versuch, alle Bereiche aufzuzählen, in denen der Computer bereits gegenwärtig ist, würde den Rahmen dieses Artikels sicher sprengen.

Der massive Einsatz von Computern hat eine scharfe Auseinandersetzung in der Gesellschaft, zwischen traditioneller und progressiver Weltanschauung, ausgelöst. Einer der sensibelsten und kritischsten Bereiche ist die Bildung bzw. der Ausbildungsbereich, der zunehmend mit den technischen Erneuerungen konfrontiert wird. Die moderne Computertechnik erfordert Kenntnisse, die heutige Lehrkräfte nicht immer oder gar nicht besitzen und schon gar nicht weitervermitteln können. Ausnahmen seien hierbei lobend erwähnt. Es waren schon immer mehr die jüngeren Generationen, die mit ihrer noch frischen Aufnahmefähigkeit und Energie Träger all dieser technischen Erneuerung waren.

Die Unterwanderung der Computertechnik im Bildungswesen stellt die traditionellen Bildungsinhalte in Frage und erfordert ihre totale Neugestaltung. Mutige Versuche, Lehrpläne zu modernisieren, scheitern an der bürokratischen Struktur stattlicher Institutionen.



Studien haben ergeben, daß das Erlernen der vier Grundrechenarten ungefähr 300 Stunden dauert. Wenn man eine Schulstunde mit einer Mark veranschlagt, kann man für diese Ausbildung etwa dreihundert Mark rechnen. Ein Taschenrechner, der sogar mehr als die vier Grundrechenoperationen beherrscht, kostet heutzutage etwa zehn Mark. Die Bedienung eines solchen Rechners ist zudem einfacher und nicht so abstrakt wie das Erlernen der fundamentalen Mathematik. Natürlich ist es nicht erstrebenswert, Lehrkörper oder Kopfrechnen abzuschaffen. Wie will man denn z. B. eine animierte Computergrafik ohne entsprechende mathematische Kenntnisse auf den Bildschirm bringen? Es geht vielmehr um die Frage, welche Inhalte und Lernziele das Bildungswesen im heutigen Computeralltag haben soll. Lernen besteht mit Sicherheit nicht nur aus Beherrschung von Lösungswegen, genauso wie man allgemeine Bildung nicht mit der Automatisierung denkender Fähigkeiten gleichsetzen kann. Aber weil in unserem Bildungssystem häufig diese Automatisierung im Vordergrund steht, fragt man sich, ob nicht der zukünftige Name der Lehrer 'Mr. Robot' lautet.

Marcelo Merino

Inhalt

Allgemeines

Software

Hardware

Grundlagen

Kurse

Aktuelles





Metamorphose bei der ST

So wie sich eine Raupe in einen Schmetterling verwandelt, so hat sich die ST Computer Redaktion GbR in die **„Merlin" Computer GmbH** verwandelt. Den notwendigen Erweiterungen unseres Marktspektrums (Hardware- und Softwareangebot u. a.) wurde damit Rechnung getragen. Auch telefonisch werden wir uns in Zukunft natürlich mit dem neuen Namen melden, also nicht wundern. Die neue alte Adresse lautet:

„Merlin" Computer GmbH
Industriestr. 26
6236 Eschborn
Tel.: 0 61 96 / 48 18 11
Fax: 0 61 96 / 4 11 37

MERLIN computer gmbh

Monitor-Umschalter für den ST

Ein Monitor-Umschalter, der endgültig Schluß mit dem ewigen Umstecken beim Anschluß von Schwarzweiß- und Farbmonitoren macht, bietet die in Dortmund ansässige Firma Kiesenberg. Das Gerät wird über einen 13-poligen Monitorstecker mit dem ST verbunden. Ein eingebauter Schalter schaltet alle notwendigen Signale um. Der Verkaufspreis des Umschalters (MTV-1) beträgt DM 68,–.

Kiesenberg Verlag
Postfach 579
4600 Dortmund 1
Tel.: 02 31 / 57 74 62

Das GOMICO FILE-SWITCH Programm

Der Umstand, daß im Desktop nur Platz für sechs Accessory-Programme ist und daß für verschiedene Anwendungsbereiche die verschiedensten Programme im AUTO-Ordner benötigt werden, macht entweder das Anlegen von verschiedenen Boot-Disketten nötig, wobei die gerade gewünschte Kombination garantiert nicht verfügbar ist, oder man kopiert sich die gerade benötigten Files auf die entsprechende Diskette, was besonders mit nur einem Laufwerk zeitaufwendig und umständlich ist.

Mit dem FILE-SWITCH Programm, der Firma Goldbach & Milksch, ist es jetzt möglich bis zu 20 ACC's und bis zu 20 Programme im AUTO-Ordner zu haben. Das Programm kostet DM 29,–.

Goldbach & Miksch
Computerprogramme
Bahnhofstr. 2–4
8782 Karlstadt 2

Ein Tastaturgehäuse für den ST

Seit neuestem wird ein Tastaturgehäuse für die ATARI ST Modelle 260 bis 1040 von Rhothron angeboten. Das Gehäuse eignet sich insbesondere zur Unterbringung der Tastatur nach einem Umbau des Motherboards in ein PC-Gehäuse. Das Tastaturgehäuse bietet nur Platz für die reine ATARI-Tastatur und bietet eine elegante Lösung, sich des etwas zu großen ATARI-Gehäuse zu entledigen. Der Preis beträgt DM 148,– und beinhaltet das Gehäuse mit Befestigungsmaterial. Ein Tastaturkabel kann zum Preis von DM 25,– ebenfalls bei der gleichen Firma bezogen werden.

Rhothron GmbH
Tiergartenstraße 5–7
6650 Homburg/Saar
Tel.: 0 68 41 / 7 18 05

Bilder via Scanner

Ein Modul, das das Lesen von Grafikvorlagen ermöglicht, bietet die Firma Schlegel Datentechnik an. Mit Hilfe eines gewöhnlichen Druckers können beliebige Bildvorlagen in den Speicher des Rechners eingelesen, anschließend abgespeichert und mit einem Mal- oder Zeichenprogramm weiterbearbeitet werden.

Der Scanner arbeitet mit einem hochauflösenden Reflexgeber von Hewlett-Packard. In der Handhabung ist das Gerät sehr einfach gehalten und ein Eingriff in den Rechner nicht notwendig. Das Modul wird nur in den ROM-Port des ATARI eingesteckt. Der ST SCANNER, wie sich das Gerät nennt, kostet DM 199,–.

Schlegel Datentechnik
Schwarzachstraße 3
7940 Riedligen
Tel.: 0 73 71 / 23 17



Neues von Creative Computer Design (CCD)

„Wir befinden uns in keinem Winterschlaf und sind auch nicht auf die Bahamas umgezogen. Wir arbeiten!!". In einem so lockeren Tonfall erklärt Dirk Beyelstein von CCD das lange Warten auf das neue Update von ST Pascal plus. Verschiedene Umstände haben dazu beigetragen, daß bis heute die Version 2.0 noch nicht marktreif sein konnte.

Der aktuelle Stand der Dinge ist (Anfang August): Der Compiler wird gerade in einer Pre-Version 1.97 getestet, ebenso der neue Pascal-Manager. Die Dokumentation wird im Moment aktualisiert. Man rechnet damit das Mitte August die neue Version von Pascal Plus erhältlich sein wird.

CCD
Burgstr. 9
6228 Eltville
Tel.: 0 61 23 / 16 38


Was die Amiga und der ST bieten, sollte die Software voll ausnutzen:
PSYGNOSIS
BARBARIAN+TERRORPODS
Zwei neue Spiele von Psygnosis. Ein Label, das sich ausschließlich dem Atari ST und der Amiga verschrieben hat. Kompromißlos.
BARBARIAN
Es muß schnell gehen... Atemlos kämpfe ich mich durch das Unterholz, und der dichte Wald kann mich nicht aufhalten. Da!! Ein weites Sumpfgebiet liegt vor mir. Das muß es sein – ja, es ist der Eingang zu Necrons unterirdischem Reich. Jetzt beginnt der Kampf ums Überleben, und das erste Mal fühle ich wirkliche Angst. Doch die Worte meines Vaters geben mir Mut. Ich muß es schaffen! Ich muß Necron besiegen und die Krone des Königreiches erlangen, um mein Volk zu retten.
Die Aufgabe ist furchterregend. Werde ich mich gegen die wütenden Angriffe der grausamen Monster und Drachen wehren können? Schaffe ich es, Necron überlegen zu sein?
Vertraut mir – ihr kennt mich! Ich bin Hegor, der mutigste Barbar, den die Welt je gesehen hat!
512 k Farbversion für Atari ST und Amiga – nur für Farbmonitore.
TERRORPODS
Eigentlich ist es ein schöner Anblick ... doch sobald die Sonne am Horizont verschwindet, wirkt alles grau und öde. Durch den Einbruch der Nacht und die plötzliche Eiseskälte erscheint mir die Gegend feindselig, und in der kühlen Einrichtung des D.S.V. fühle ich mich fast wie zu Hause.
Meine melancholische Stimmung wird durch schrilles Sirenengeheul gestört. Die Anzeigeninstrumente spielen verrückt, die vielen Lichtblitze werden unkontrollierbar. Tausend Gedanken schießen mir durch den Kopf. Was ist passiert?
Wie wild springe ich auf und renne zum Kommando-Terminal, arbeite mich durch eine unüberschaubare Masse an Informationen, um den Fehler zu finden.
Das Grauen packt mich – es kann nicht sein! Die Terrorpods!
512 k Farbversion für Atari ST und Amiga – nur für Farbmonitore.
Wer wissen will, was wir außer Barbarian+Terrorpods noch für tolle Spiele haben, dem schicken wir gerne Prospekte zu.
Name
Straße
PLZ
Ort
An: ariolasoft, Carl-Bertelsmann-Str. 161, 4830 Gütersloh.
ariolasoft
Ein Unternehmen der Bertelsmann AG
68000er 8/87
SEEING IS BELIEVING

Massenspeicher für ATARI ST

Eine Serie von Festplatten und Streamern unterschiedlicher Kapazität, am ATARI ST anschlußfertig, bietet die Firma PMD aus Vaterstetten an. Festplatten von 20 bis 70 MB sowie Streamer von 20 und 60 MB sind im Moment erhältlich. Alle Festplatten können in acht Partitions unterteilt werden, wobei jeder Partition eine beliebige Größe zugeteilt werden kann. Da alle Festplatten über eine SCSI-Schnittstelle angeschlossen werden können, kann das Verbindungskabel bis zu zwei Meter lang sein.

Die anschlußfertigen Geräte sind, werden mit eingebautem Netzteil und Durchschleifstecker für den DMA-Anschluß des ST geliefert. Da die Festplatte sowie der Streamer mit einem SASI-Interface betrieben werden, kann man bis zu acht Geräte am DMA-Kontroller des ATARI ST anschließen.

Durch die mitgelieferte Software kann die Festplatte formatiert bzw. in verschiedene Partitionen eingeteilt werden. Die Größe der einzelnen Partitionen ist in Sektoren zu je 512 Bytes frei wählbar und beträgt maximal 16 Megabytes. Ebenso zum Lieferumfang gehört ein anderes Programm, um die Festplatte bei Bedarf auf Fehler zu überprüfen.

Sowohl bei Festplatte als auch bei Streamer konnte dank optimierter Bauweise auf einen Lüfter verzichtet werden. Der Streamer erfordert als Speichermedium Datencassetten mit 3,81 mm breitem Magnetband (ähnlich den Tonbandcassetten). Die Bedienung erfolgt durch ein Accessory, mit dem direkt aus einem Programm heraus einzelne Dateien oder ganze Partitions auf das Band gesichert und zurückgeladen werden. Der Streamer arbeitet auch mit Festplatten anderer Hersteller zusammen. Alle Geräte werden mit einem ausführlichen deutschen Handbuch und der benötigten Software geliefert.

Die Preise für die Festplatten:
- 20 MB Festplatte DM 1498,–
- 40 MB Festplatte DM 2998,–
- 70 MB Festplatte DM 4498,–

Die Preise für die Streamer:
- 20 MB Streamer DM 1798,–
- 60 MB Streamer Dm 2298,–

Subsystem 20 MB Streamer + 20 MB Festplatte DM 3198,–.

PMD
Postfach 100105
8011 Vaterstetten
Tel.: 08106/33941

RadioFax ST

AFUSOFT RadiFax ST ist ein Programm, daß den Empfang von Telefax-Bildern, wie z. B. METEOSAT-Bilder oder Seeamtswetterkarten, ermöglicht. Das Programm simuliert einen komfortablen Telefax-Empfänger. RadioFax ST arbeitet sowohl mit dem Monochrom- als auch mit dem Farb-Monitor zusammen. Das Programm stellt sich automatisch auf den verwendeten Monitortyp ein und paßt das Bildformat der jeweiligen Auflösung an. Die Software erlaubt es empfangene Bilder, während des Empfangs, auf Diskette oder Festplatte abzuspeichern. Empfangene oder gespeicherte Bilder können bei Bedarf über einen Standard-Matrixdrucker ausgedruckt werden. RadioFax ST wird mit einer ausführlichen Bedienungsanleitung in deutscher Sprache ausgeliefert. Der Preis beträgt DM 178,–.

Buchhandlung Franke
Steiner Straße 5
Tel.: 07232/81664

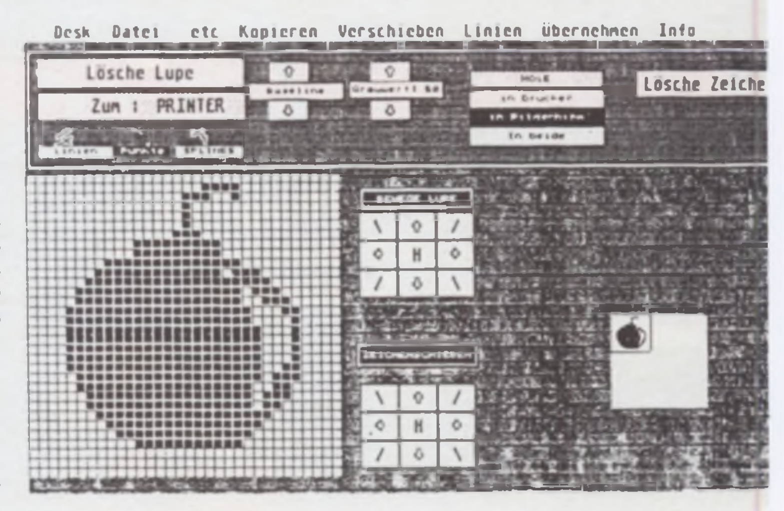
CAD Project

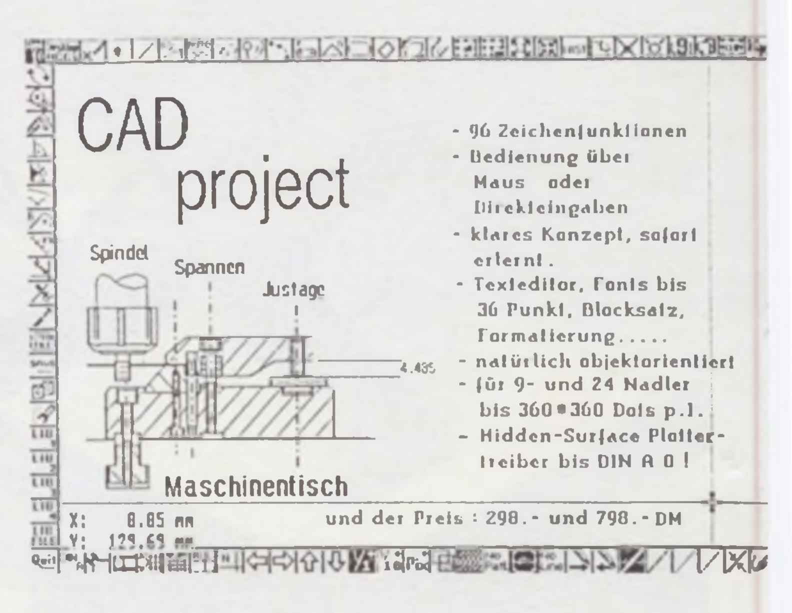
So heißt diese neue Entwicklung der Firma D. Schwarzstein Software, die speziell für die ST-Rechner von ATARI fertiggestellt wurde. Durch enge Zusammenarbeit des Entwicklerteams und CAD-Experten hat sich ein interessantes Endprodukt ergeben.

Alle Funktionen sind in Form von Icons am Monitor ständig sichtbar und leicht mit der Maus erreichbar. Neben umfangreichen Zeichen- und Konstruktionsfunktionen sind auch sämtliche Features von Desktop Publishing Programmen implementiert. Besonderer Wert wurde auch auf die Qualität der Ausgabe gerichtet, sie erfolgt wahlweise auf einen 9- oder 24-Nadel-Drucker oder einen Plotter. An einem Laserdrucker-Treiber wird im Moment noch gearbeitet.

Der Preis dieses Programms beträgt DM 798,–. Eine eingeschränkte Ausführung ist für DM 298,– zu erhalten. Eine Demo-Version zum Ausprobieren wird für DM 10,– angeboten.

Philgerma GmbH
Ungererstr. 42
8000 München 40
Tel.: 089/395551





GFA-Desktop-Publisher

Das in Düsseldorf ansässige Softwarehaus GFA Systemtechnik hat in diesem Sommer viel Neues zu melden. Ende August dieses Jahres ist der GFA-Publisher, ein nach Herstellerangaben sehr leistungsfähiges Desktop-Publishing-Programm, zu erhalten. Ein paar Features, die in diesem Programm enthalten sind:

- Vollständig integrierter Texteditor
- Einzug von Grafiken, GEM Meta-Files und GEM-Image-Files
- Grafiken können manipuliert werden
- Programminterne Auflösung unabhängig von der eingestellten Bildschirm- und Drucker-Auflösung 2540 Dots/Inch.
- Deutsche Silbentrennung
- Voll GEM-gesteuert

Der Verkaufspreis beträgt DM 398,–.

Ferner gibt es jetzt eine neue Version des GFA-BASIC, das sogenannte GFA-BASIC 68881, das den Motorola Floating-Point-Coprozessor 68881 völlig unterstützt. Das Programm wird in Form eines kompletten Entwicklungssystempakets, das den neuen Interpreter und Compiler beinhaltet, ausgeliefert. Diese Version läuft nur wenn der oben genannte Coprozessor in den Rechner eingebaut ist. Der Verkaufspreis dieses Programmpakets beträgt DM 349,–. Ebenfalls wurde eine neue Version 2.01 des GFA-BASIC Interpreters bekannt gegeben. Der Compiler wurde dementsprechend angepaßt.

GFA-Systemtechnik GmbH
Postfach 190263
4000 Düsseldorf 11
Tel.: 0211/588011



Monitor-Umschalter

Ein weiterer Monitor-Umschalter wird von der Firma PCP angeboten. Es handelt sich dabei um einen Hardware-Umschalter, der es ermöglicht, sowohl Monochrom- als auch Farbmonitore gleichzeitig anzuschließen. Das lästige Umstecken der beiden Anschlußkabel entfällt. Eine gleichzeitige Ausgabe der Bildinformation auf beiden Monitoren ist allerdings aus systembedingten Gründen nicht möglich. Zusätzlich wurden noch zwei Cinch-Buchsen eingebaut, über die mit einem einfachen Verbindungskabel der Sound auf einer HiFi-Anlage ausgegeben werden kann. Das Gerät wird einfach in die Monitor-Buchse des ST gesteckt. Der Preis für den Monitor-Umschalter beträgt DM 59,–.

PCP
Am Hohen Stein 36
6200 Wiesbaden
Tel.: 06121/65758



Mainradio und ST-Computer verlosen Jahresabos und Spiele

Freuen können sich alle ST- und Amiga-Besitzer, die im Bereich des Senders Mainradio (WÜrzburg und ca. 60 km Umgebung) wohnen. Bereits im Juni und Juli wurden beim Mainradio in Zusammenarbeit mit unserer Redaktion mehrere Jahresabos von Kickstart und ST-Computer verlost.

Im August und September gibt es wieder einige Abos und dazu noch Programme von Psygnosis (Barbarian), Rainbow Arts (Bad Cat), Application Systems u. a. m. zu gewinnen. Mainradio findet man auf UKW, Frequenz 103. Die Verlosung findet in der Morgenshow (5.30 bis 7.30 Uhr) statt.

MAINRADIO
Augustinerstr. 15
8700 Würzburg
Tel.: 0931/50881





Floating-Point-Coprozessor 68881



Die Firma Lischka Datentechnik bietet einen Mathematik-Coprozessor unter der Bezeichnung LD-FPU für den ATARI ST an. Der benutzte Floating-Point-Prozessor 68881 von Motorola ist mit 16 MHz getaktet (höhere Taktraten sind auf Wunsch lieferbar) und wird mitsamt einer kleinen Platine, die die nötige Hardware enthält, auf den 68000er Prozessor aufgelötet bzw., falls der 68000er gesockelt ist, zwischengesteckt. Durch diesen Hardwarezusatz wird die Rechenleistung des ST enorm gesteigert. Die von ATARI bekanntgegebene Original-Adresse für Coprozessoren dieser Art wurde bereits berücksichtigt, wodurch die LD-FPU kompatibel zu dem in Zukunft erscheinenden ATARI-Mathematik-Coprozessor ist. Auch auf die Blitter-Adressierung wurde Rücksicht genommen.

Mitgeliefert wird ein ausführliches deutsches Handbuch mit einer Kurzanleitung zum Programmieren, das Original-Motorola-Handbuch zum 68881 und eine Diskette, auf der sich mehrere Beispielprogramme, eine Lattice- und eine Megamax C-Anpassung befinden. Ferner gibt es eine Anpassung des GFA BASICs (Original-GFA BASIC 68881 Interpreter und Compiler) gegen einen Aufpreis von DM 169,– zu beziehen. Demnächst sind weitere Anpassunger. für Mark Williams C, Pro-Fortran 77, Pascal, Modula II, Forth und Omikron BASIC erhältlich. Der Preis für den Coprozessor beträgt DM 729,–

Lischka Datentechnik
Hochstr. 22
4173 Kerken 2 / Aldekerk
Tel.: 02833/7388

OMIKRON Basic mit 68881

Für den OMIKRON Basic Interpreter ist ab sofort auch eine Version erhältlich, die den Fließkommaprozessor 68881 unterstützt. Dadurch werden alle mathematischen, im besonderen die doppeltgenauen Operationen wesentlich beschleunigt.

Der Compiler erhält eine zusätzliche Bibliothek und läuft damit, sowohl mit, als auch ohne den Arithmetikprozessor, je nachdem ob dieser eingebaut ist. Von Interpreter und Compiler sind Updates erhältlich.

OMIKRON Software
Erlachstr. 15
7534 Birkenfeld
Tel.: 07082/5386

METAFONT für ST TEX

METAFONT ist eine Art Programmiersprache, womit sowohl Zeichensätze als auch kleine Grafiken entworfen werden können, um sie später in dem Textformatiersystem TEX zu verwenden. METAFONT ist kein interaktiver Fonteditor, mit dem man mittels eines graphischen Eingabegeräts (Maus, Graphiktablett, etc.) die Zeichen eines Fonts in Dialogform am Bildschirm entwirft. Bei METAFONT wird für jeden Buchstaben ein Programm geschrieben, das später in die gewünschten Zeichen übersetzt wird. Dieses Konzept von METAFONT scheint zunächst einmal sehr viel umständlicher und komplizierter zu sein als eine interaktive Lösung, aber solche Programme besitzen den Vorteil, daß sie mit Parametern versehen werden können. Typische Parameter sind etwa:

- Die Höhe und Breite der einzelnen Zeichen eines Zeichensatzes
- Die Breiten einzelner Linienzüge eines Zeichens
- Transformation der Zeichen
- Die Eigenschaften des Ausgabegeräts (Auflösung)

ST-METAFONT kostet DM 298,–.

TOOLS GmbH
Kaiserstr. 48
5300 Bonn 1
Tel.: 0228/229791

ATARI Messe in Düsseldorf
Eine Messe zum Anfassen

Vom 18. bis 20. September veranstaltet ATARI eine Ausstellung auf dem Messegelände in Düsseldorf. Verschiedene Anbieter von Software- wie Hardwareprodukten für alle ATARI Rechner aus Deutschland und dem Ausland werden sich auf der über 2.500 qm großen Ausstellfläche treffen.

Die gesamte ATARI-Produktpalette, von Schulsoftware bis Künstliche Intelligenz oder Desktop-Publisher wird zu sehen sein. In verschiedenen Veranstaltungen werden die Produkte ausdiskutiert und erklärt.

Für alle Musik-Freunde stellt ATARI einen Sonderraum, so ständig Midi-Vorführungen stattfinden werden. Renommierte Programme wie 'Twenty-Four' von Steinberg Research oder 'MidiTrack ST Profesional' von Hybrid Arts und viele andere (über dreißig Produkte dieser Klassen sind auf dem deutschen Markt erhältlich) werden dauernd vorgeführt. Produktlösungen für den kaufmännischen Bereich werden mit ihrem gesamten Spektrum vertreten sein. Auch die starken Grafikfähigkeiten des ATARI ST wird man noch einmal bei der Vorführung der vertretenen Mal- und Zeichenprogramme bestaunen können.

Für alle eingefleischten ATARI-Fans wird aus den ATARI Shop jed Menge Zusatzartikel für wenig Geld zu haben sein. Kulis, Regenschirme, Aufkleber, etc. alles mit dem ATARI Symbol versehen, kann man mit nach Hause nehmen. Damit sich niemand hoffnungslos auf dem über 100 Aussteller umfassenden Ausstellungsgelände verläuft, bietet ATARI einen Messeführer mit allen wichtigen Informationen, wie Standnummer, Name der Aussteller, Produkt, etc. Es werden auch jede Menge Spiele zu sehen sein und das nicht nur für die ST Serie, denn die 8 Bit Rechner sind ebenso vertreten. ATARI selbst wird seine neuen Produkte vorführen. Die Mega-Modelle (2 und 4 Megabyte-Rechner), der Laserdrucker sowie der ATARI-PC werden jedem zugänglich sein.

Die ST Redaktion wird bei diesem wichtigen Ereignis auch mit einem eigenen Stand vertreten sein.

Programmserie für kaufmännische Anwender

Eine Serie von fünf Programmodulen für kaufmännische Anwendung werden von der Gesellschaft für Finanz- und EDV-Beratung angeboten. Sie wird zur Systems in München vom Sybex-Verlag vorgestellt. Die Programme sind einzeln oder als Paket zu beziehen und werden mit umfangreichem Handbuch geliefert. Jetzt eine kurze Beschreibung aller fünf Module:

Tos-Manager als Hilfe für den Einsteiger und den Fortgeschrittenen gedacht, enthält eine Ramdisk, Formatier- und Kopierroutinen und einige Accessories. Über dieses Modul ergibt sich die Möglichkeit die anderen Module der Serie aufzurufen. Der Preis beträgt DM 98,–.

Adress-/Kundenverwaltung mit sehr umfangreichen Datenbearbeitungsmöglichkeiten. Datenspeicherung, Selektions-, Suchfunktionen, Zahlungsverkehrabwicklung, Serientelex, etc. sind alles Funktionen die selbstverständlich eingebaut sind. Der Preis beträgt DM 149,–.

Lagerverwaltung/Fakturierung Stücklisten, Kassenabrechnung, Bestandsbuchführung, Verbrauchsfolgeannahme für Bestandsbewertung wird unter anderem in einem Modul angeboten. Der Preis beträgt DM 398,–.

Finanzbuchhaltung, ein nach dem europaweit gültigen Vier Bilanzrichtliniengesetz Buchhaltungsprogramm. Dieses Modul ermöglicht die Verarbeitung von Sammelbelegen, DATEV-Kontenrahmen, Mandanten, Anlagen und Abschreibung. Preis DM 498,–

Lohn/Gehalt ein Modul, das den Zahlungsverkehr, Formularwesen, Zeitkonten bis hin zur Lohnpfändung von allein erledigt.`Preis DM 198,–.

Gesellschaft für Finanz und EDV-Beratung
Königsteiner Straße 76
6232 Bad Soden
Tel.: 06196/62139

Buchhaltungsreihe fibuMAN

Mit fibuMAN Version 1.2 ist ein neues Buchhaltungspaket der Firma PRODATA auf den Markt gekommen. Es ist voll GEM-eingebunden. Ein umfangreiches Angebot an jederzeit einsehbaren übersichtlichen Listen und Auswertungen ist jederzeit auf Bildschirm, Drucker oder Diskette auszugeben. Der Einfachheit halber lassen sich alle relevanten Listen als Sammeldruck am Ende des Monats ausgeben. Viele anwenderfreundliche Features sind eingebunden, so weist fibuMAN z. B. auf unsinnige Buchungen hin oder zeigt vor und nach jeder Buchung die aktuellen Salden der angesprochenen Konten. Ferner lassen sich monatlich wiederholende Buchungen nach einmaliger Eingabe automatisch buchen. Während eines Buchungsvorgangs lassen sich neue Konten anlegen.

Alle fibuMAN-Programme besitzen Schnittstellen zu einem beliebigen Programm (z. B. Textverarbeitung), zu fibuSTAT, einem ausführlichen Statistikmodul, zu faktuMAN, einer komfortablen Faktura.

Folgende fibuMAN-Versionen sind erhältlich:

fibuMAN e (Einnahme-Überschuß-Rechnung) DM 345,–, **fibuMAN f** (Finanzbuchhaltung) DM 645,–, **fibuMAN m** (mandantenfähige Fibu) DM 845,–.

Alle Produkte werden von PRODATA mit ausführlichem deutschen Handbuch und einer Checkliste als Kurzinfo ausgeliefert. Ein Aufstieg in der fibuMAN-Reihe ist jederzeit unter Anrechnung des aktuellen Preises der ursprünglichen Version möglich. Für Besitzer der fibuMAN f-Version wird die in fibuMAN m eingeschlossene BWA zum Preis von DM 95,– angeboten.

PRODATE bietet prompten Update-Service, kostenlose Hotline, Unterstützung (täglich 10–23 Uhr) und 24-Stunden-Lieferung.

PRODATA
Broicherstr. 39
5060 Bergisch Gladbach 1
Tel.: 02204/51456

DER DRUCKER

Stellen Sie sich einmal folgende Situation vor. Sie verfügen über einen leistungsfähigen Computer. Des weiteren sind Sie im Besitz einer vorbildlich und hundertprozentig nach Ihren Wünschen funktionierenden Software. Die Ergebnisse auf dem Bildschirm sehen ebenfalls noch erstklassig aus. Wenn es dann aber darum geht, Druck zu machen, fehlen Ihnen die Worte. Denn Ihr Drucker kann leistungsmäßig nicht mithalten und bietet eine blasse Vorstellung. Das kommt Ihnen bekannt vor? Dann gibt es nur eines. Schwenken Sie um und stellen Sie Ihrem Computer einen gleichwertigen Partner an die Seite. Zum Beispiel den NEC Pinwriter P6 Color. 24 Nadeln garantieren perfekte Druckergebnisse in schwarzweiß und Farbe.



Pinwriter P6 Color

**NEC Pinwriter P6 Color.
Ein vorbildlicher Partner für Ihren Computer.**

Grafiken, Text und EDV-Listen hinterlassen jederzeit den besten Eindruck. Und selbst komplexe Desktop Publishing- oder CAD-Aufgaben sind für einen Drukker wie den NEC Pinwriter P6 Color eine leichte Übung. Vom günstigen Preis einmal ganz zu schweigen.

NEC Pinwriter P6 Color
- **24-Nadel-Drucktechnologie**
- **Druckgeschwindigkeit: max. 216 Zeichen/Sek.**
- **Schreibbreite: 80 Zeichen/Zeile**
- **Auflösung: 360 x 360 Punkte/Zoll**
- **Betriebsgeräusch: 53 dBA (Quiet Mode)**

DER TREIBER

Sie wissen es nur zu gut: Eine Kette ist so stark wie ihr schwächstes Glied. Viele Computer/Software/Drucker-Konfigurationen haben ebenfalls eine klare Schwachstelle: den Druckertreiber. Mit einem qualitativ minderwertigen Treiber werden die dem Drucker seitens Computer und Software übermittelten Informationen nur unvollkommen umgesetzt und zu Papier gebracht. Die Ergebnisse sind dann naturgemäß enttäuschend. NEC läßt es nicht dazu kommen. Denn die eigenentwickelten bzw. in enger Zusammenarbeit mit den wichtigen Softwarehäusern erstellten Druckertreiber von NEC garantieren beste Ergebnisse. Nur mit einem perfekten Druckertreiber ist die Software in der Lage, alle Leistungen des Druckers vollkommen auszunutzen. Alle NEC Druckertreiber setzen die Computerdaten für den Drucker hundertprozentig verständlich um. Welche Programme von den NEC Pinwritern erstklassig unterstützt werden, läßt sich leicht ermitteln. Denn zu diesem Zweck gibt es den NEC Software Report.

- **Ein qualitativ minderwertiger Druckertreiber beeinträchtigt den Kommunikationsfluß zwischen Computer und Drucker.**

Textprogramme

Programm-Name	P560 P565	P560XL P565XL	P660 P665	CP660 CP665	P760 P765	CP760 CP765
AGTEXT	DOS	DOS	DOS	DOS	DOS	DOS
Atari 1st Word V 106	NEC	KF	NEC	KF	NEC	KF
Easy 10	ORG	KF	ORG	KF	ORG	KF
Easywriter 2	KOM	KOM	KOM	KOM	KOM	KOM
Signum	ORG	KF	ORG	KF	ORG	KF
Euroscript V 2	ORG	ORG	ORG	ORG	ORG	ORG
Gem Write	NEC	NEC	NEC	NEC	NEC	NEC
Info-Text	ORG	KF	ORG	KF	ORG	KF
MS Window Write	KOM	KOM	KOM	KOM	KOM	KOM
MS Word V 2	ORG	ORG	ORG	ORG	ORG	ORG
MS Word V 3	ORG	ORG	ORG	ORG	ORG	ORG
Multimate 3.31	ORG	ORG	ORG	ORG	ORG	ORG
Papyrus	NEC	KF	NEC	KF	NEC	KF
PC Text 3	NEC	NEC	NEC	NEC	NEC	NEC
PC-Write	ORG	KF	ORG	KF	ORG	KF
Profi-Text	ORG	ORG	ORG	ORG	ORG	ORG
Prosa	ORG	KF	ORG	KF	ORG	KF
Rechentext	ORG	ORG	ORG	ORG	ORG	ORG
Samna Word 3	ORG	ORG	ORG	ORG	ORG	ORG
Science Text V 657	ORG	KF	ORG	KF	ORG	KF
ST Textomat	ORG	ORG	ORG	ORG	ORG	ORG
Tex Ass Window +	NEC	NEC	NEC	NEC	NEC	NEC
Textomat PC	ORG	ORG	ORG	ORG	ORG	ORG
Volkswriter	ORG	ORG	ORG	ORG	ORG	ORG
Wordperfect	NEC	NEC	NEC	NEC	NEC	NEC
Wordstar 2000 V 1.01	ORG	KF	ORG	KF	ORG	KF
Wordstar 3.4	NEC	NEC	NEC	NEC	NEC	NEC
Wordstar 3.45	ORG	ORG	ORG	ORG	ORG	ORG

Auszug aus NEC Software Report (2/87)

DIE QUALITÄT

Dies dürfte selbst dem absoluten Drucker-Neuling sofort einleuchten: 24-Nadel-Drucker sind qualitativ den 9-Nadel-Druckern überlegen, und daß Drucker mit 24-Nadel-Technologie bessere Druckergebnisse liefern, ist folglich auch keine Frage. Hier landet man zwangsläufig bei NEC.

- **Nur NEC Pinwriter, die mit dem neuen Benutzerhandbuch in Deutsch und Original-NEC-Seriennummer ausgeliefert werden, sind vollkommen in das NEC-Servicepaket integriert. Hierzu zählt u. a. 12-Monate-Garantie, Treiber-Software, telefonische Hotline und vieles mehr.**



Original NEC Bedienungshandbuch (Ringbuch, 312 Seiten)

Die 24 Nadeln der NEC Pinwriter erzeugen ein optimales Druckbild, gleichgültig ob bei Text-, Grafik-, CAD- oder Desktop Publishing-Anwendungen. Sie wissen es ja: Der Drucker ist quasi die Visitenkarte Ihres Computers. Und nur Spitzengeräte bringen die Leistung Ihres Computers in vollem Umfang zum Ausdruck.

Der neue Standard:

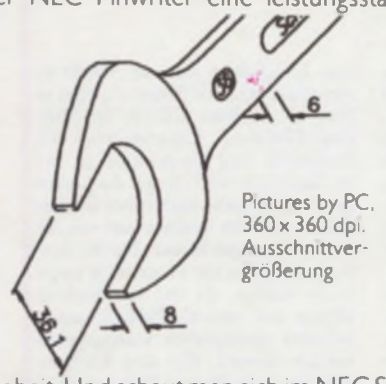
Betrachtet man einmal die Ausdruckqualität der NEC Pinwriter und vergleicht diese mit der anderer Geräte in der 24-Nadel-Klasse, wird deutlich, warum in Testberichten der Computer-Fachpresse die Leistung und Qualität der NEC Pinwriter immer häufiger als Standard angegeben wird. Diesem Urteil der Fachpresse ist nichts mehr hinzuzufügen.

NEC PINWRITER UND IHR PERSONAL COMPUTER

NEC Pinwriter sind die vorbildliche Ergänzung zu Ihrem Personal Computer. Dabei spielt es keine Rolle, ob Sie sich vornehmlich mit Text, CAD oder Grafik

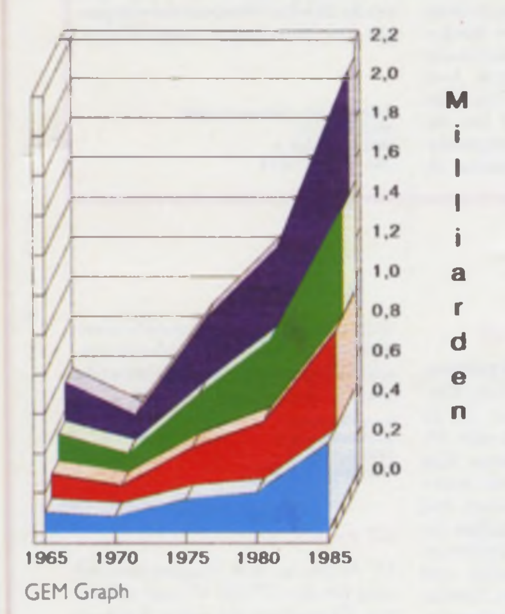
NEC Pinwriter und Ihr Personal Computer: Ein Gespann, das es in sich hat.

beschäftigen. Durch die perfekte Anpassung bilden Ihr Personal Computer und der NEC Pinwriter eine leistungsstarke



Pictures by PC, 360 x 360 dpi. Ausschnittvergrößerung

Einheit. Und schaut man sich im NEC Software Report einmal die umfangreichen Software-Lösungen an, die die 24-Nadel-



GEM Graph

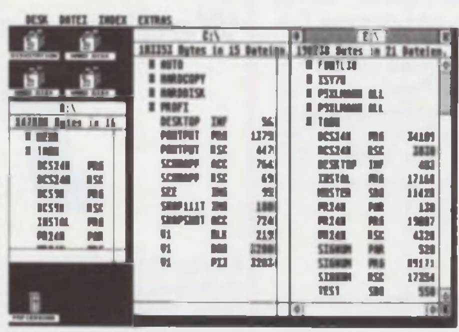
Technologie der NEC Drucker bis ins Kleinste ausnutzen, wird deutlich, warum immer mehr Personal Computer-User sich für einen NEC Pinwriter entscheiden.

Jede dieser Schriftarten gibt doppelter oder doppelter Höhe, Fettschrift, Tiefstellung. Mit entsprechende SCHREIBEN IN VERSALIEN

Auszug MS Word 3,0

NEC PINWRITER UND IHR ATARI ST

Atari ST-Anwender sind von der Grafikleistung der NEC Pinwriter tief beeindruckt. Kein Wunder, bringen die NEC Pinwriter doch die Bildschirminformationen durch die NEC Grafik-Routine erst-



Atari Hardcopy

klassig zu Papier. Atari ST-User wollen natürlich einen perfekten Drucker, der, ebenso wie Ihr Computer, jederzeit ein-

4.1 Amine

Amine sind stickstoffhaltige organische Verbindungen, die als Alkylderivate des Ammoniaks betrachtet werden können. Nach der Zahl, der im NH_3-Molekül durch Alkylgruppen ersetzten Wasserstoffatome, unterscheidet man zwischen *primären, sekundären* und *tertiären Aminen*. Die Namen bestehen aus den Bezeichnungen für alle an den Stickstoff gebundenen Alkylgruppen und dem Wort -amin. Die Alkylgruppen werden nach steigender C-Zahl aufgezählt.

CH_3-NH_2	CH_3-NH-CH_3	CH_3-N(CH_3)-CH_3
Methylamin *primäres Amin*	**Dimethylamin** *sekundäres Amin*	**Trimethylamin** *tertiäres Amin*

CH_3-NH-CH(CH_3)$_2$	CH_3-CH_2-N(CH_3)-C(CH_3)$_2$-CH_3
Methylisopropylamin	**Methyläthyltertiärbutylamin**

Signum!

fach zu bedienen ist. Wenn auf dem Atari ST Programme wie 1. Word oder Signum! im Textbereich, Profi Painter für den „Grafiker" bzw. STAD, GFA-Draft oder

Einfach professionell: Atari ST und NEC Pinwriter.



Profi Painter

CAMPUS für CAD-Anwendungen laufen, weiß man die Qualitäten der NEC Pinwriter zu schätzen.

NEC PINWRITER UND IHR COMMODORE AMIGA

Die Fähigkeiten des Commodore Amiga liegen unbestritten im Anwendungsgebiet Grafik. Aber was auf dem Bildschirm leuchtend und kristallklar auf-



taucht, soll natürlich auch auf Papier perfekt aussehen. Und so mancher Drucker verliert hier plötzlich alle Farbe. Deshalb ist allen Commodore Amiga-Anwendern der NEC Pinwriter P6 Color zu empfehlen. Er gibt Grafiken nuancenreich und detailgenau wieder. Bei



einer Auflösung von 360 x 360 Punkte/Zoll nicht weiter verwunderlich. Da die

Ideales Paar: Ihr Commodore Amiga und der NEC Pinwriter P6 Color.



24 Nadeln einzeln adressierbar sind, lassen sich selbst hochaufgelöste Bilder perfekt ausdrucken, wie die drei Printouts mit Deluxe Paint beweisen.

NEC

NEC Deutschland GmbH

Dr. G. Dohrenberg 1000 Berlin 30 Tel.: 030/2 19 00 70
CHS 2000 Hamburg 1 Tel.: 040/23 22 23
L + S 3000 Hannover 81 Tel.: 05 11/8 42 44-0
SYSDAT 5000 Köln 40 Tel.: 02 21/4 89 05-0
S.E.H. 6455 Erlensee Tel.: 0 61 83/8 30
Magirus 7022 Leinfelden-Echterdingen Tel.: 07 11/7 59 04-0
SCHWIND 8033 Martinsried Tel.: 089/8 57 20 47
SYSDAT CH-3627 Heimberg Tel.: 033/37 70 40
INDUTRONIC A-9020 Klagenfurt Tel.: 0 42 22/4 36 93-0

Neue 32-Bit-VME Systeme von MOTOROLA

Mit dem **MVMESYS133V02** kündigt MOTOROLA ein neues, preisgünstiges 32-Bit-VME-System an, dessen Grundlage ein 32-Bit-Mikroprozessor vom Typ MC68020 und ein Gleitkomma-Coprozessor MC68881 bilden. Das System läuft unter dem Betriebssystem VERSAdos, das echtzeitfähig ist und sich für Multiuser- und Multitasking-Betrieb eignet. Sowohl für die Entwicklungsphase als auch für das Zielsystem bildet dieses multitasking- und multiuserfähige Echtzeitbetriebssystem ein denkbar günstiges Betriebsumfeld.
Standardmäßig ist das MVME-System mit zwei seriellen RS-232-Schnittstellen ausgestattet. Sollten zusätzliche serielle Leitungen benötigt werden, stehen verschiedene Schnittstellen-Module zur Auswahl.

Zum Umfang des Softwarepakets gehören ein bildschirmorientierter Editor, Assembler für die MC68000-Familie, ein Linker und Gerätetreiber, die als Quell- und als Objektcode für sämtliche VME-Module zur Verfügung stehen.

Separat werden auch höhere Programmiersprachen wie Pascal, Fortran und C angeboten.

Motorola GmbH
Arabellastr. 17
8000 München 81
Tel.: 0 89 / 92 72-0

Neue Spiele von Rainbow Arts

Kurz nach Redaktionsschluß erreichten uns noch einige „heiße News" von Rainbow Arts. Wir mußten ein bißchen zaubern, deshalb bitten wir um Verständnis, daß die Information darüber nur relativ kurz ausfällt:

Street Gang:

Mickys Leben war bisher ebenso normal wie langweilig. Mit seinen Eltern wohnte er in einer Kleinstadt auf dem Land. Doch dann kam der Tag, an dem sich alles ändern sollte: Seine Eltern zogen mit ihm nach New York. Dort werden die Straßen von den verschiedensten Banden regiert. Um Mitglied in der Bande seiner Straße zu werden, macht sich Micky auf den Weg, dem berüchtigten Locke sein heißgeliebtes Haarbüschel zu rauben. Mickys Abenteuer ist ein Geschicklichkeitsspiel, die Anleitung dazu besteht aus einem kurzen Comic.



In achtzig Tagen um die Welt

Die Faszination einer Weltreise, der Reiz fremder Kulturen und die Erforschung unheimlicher Plätze hat schon Jules Verne zu dem gleichnamigen Weltbestseller inspiriert. In einem edlen englischen Club schließt sein Held Philias Fogg im Jahre 1872 die Wette ab, die Welt in 80 Tagen zu umrunden. Diese Wette, von den Clubmitgliedern belächelt, entwickelt sich zu einem Abenteuer auf Leben und Tod. Der Spieler übernimmt die Rolle von Passpartout, dem treuen Diener. Um seinen Herrn vor Problemen zu bewahren, befreit er gefangene Jungfrauen aus heidnischen Tempeln, tanzt mit Indianern u.a.m. Das knappe Geld wird durch Kartenspielen aufgebessert.

Bad Cat:

Los Angeles 1984. Die Stadt rüstet sich für die Olympischen Sommerspiele. Große Leute halten ebensogroße Reden. Alles ist bereit für die Besuchermassen. Am Rande dieses Happenings finden sich auch die streunenden Großstadtkatzen ein, um ihre eigene Olympiade zu veranstalten. Sie treffen sich überall in der Stadt, um die ausgefallensten Wettbewerbe auszutragen. Der Favorit ist BAD CAT – wird er sich seines Namens als würdig erweisen? Für Katzen ist schon das Erreichen der Schauplätze ein nicht zu unterschätzendes Problem. In rasanten Verfolgungsjagden, stets auf der Flucht vor Hunden, muß zunächst der große Park erreicht werden. Weiter geht es dann in der übelsten Spelunke der Stadt. Beim Kegeln trifft BAD CAT auf die dicke Bulldoge aus dem Westteil der Stadt. Wird er von einer Kugel getroffen, muß er ein großes Bier austrinken. In mehr oder minder ramponiertem Zustand geht es danach ins Kanalsystem von Los Angeles. Geschicklichkeit und Mut sind hier besonders gefragt. Viele angsteinflößende und gefährliche Tiere führen im Kanal ein finsteres Dasein. Endlich im Stadion angelangt, ist sportliche Akrobatik Trumpf: Absprung, Salto und butterweiche Landung – oder ein Sturz ins kalte Wasser, das ist die Frage. Graphisch hebt sich BAD CAT angenehm vom Durchschnitt ab, auch die Titelmusik ist sehr ansprechend. Besonders gut haben uns die vielen, ganz unterschiedlichen Spielsituationen gefallen, in die BAD CAT gerät. Zu der auch den Amiga schon überschwemmenden Welle von Ballerspielen setzt BAD CAT einen angenehmen Kontrapunkt – Geschicklichkeit ist Trumpf.





Rainbow Arts & ST Computer Wettbewerb



Rainbow Arts und ST-Computer-Magazin verlosen gemeinsam 20 Originale des neuen Atari-ST-Spiels „BAD CAT". Als Sonderpreis gibt es darüberhinaus noch einen Besuch bei Rainbow Arts zu gewinnen. Um an der Verlosung der Spiele teilzunehmen, müßt Ihr nur eine Postkarte ausfüllen, „Verlosung Bad Cat" draufschreiben und die Karte an uns einschicken. Der Sonderpreis – eine Woche bei Rainbow Arts – wird unter all denjenigen verlost, die zusammen mit der Postkarte noch ein kleines selbstgeschriebenes Public-Domain-Spiel für den Atari ST einschicken. Wir erwarten keine diskettenlangen Programme; es genügt ein nettes, KLEINES Spielchen, das Ihr vielleicht einmal so ganz nebenbei geschrieben habt. Wie gesagt, besteht der Sonderpreis aus einer Einladung zu einem einwöchigen Besuch bei Rainbow Arts. Der Gewinner kann sich Tips zum Programmieren holen, darf bei der Entwicklung der neuen Programme ein Auge riskieren und bekommt einen kompletten Satz der Rainbow-Arts-Titel mit nach Hause – „wenn er brav ist", so Marc Ullrich, der RBA-Geschäftsführer. Einsendeschluß für Postkarten und Programme ist der 26. September 1987. Es gilt das Datum des Poststempels. Mitarbeiter der Redaktion und des Verlages sind von der Verlosung ausgeschlossen. Für den Sonderpreis dürfen nur Programme eingereicht werden, die frei von Rechten Dritter sind und als Public-Domain-Software unentgeltlich weitergegeben werden können. Für eine Rücksendung der eingeschickten Disketten übernehmen wir keine Garantie. Mit der Einsendung erkennt der Teilnehmer diese Bedingungen an. Im übrigen gelten die üblichen Wettbewerbsbedingungen. Der Rechtsweg ist ausgeschlossen.

Rainbow Arts Software GmbH
Münsterstr. 27
4830 Gütersloh 1
Tel.: 0 52 41 / 1 68 88

Patienten-Diagnose mit ST

Unter dem Namen Privatliquidation für Ärzte bietet die Firma Kray-Electronic ein Programmpaket an, das die Verwaltung einer Arztpraxis oder -klinik erleichtert. Das Programm läuft seit Oktober 1986 und wurde inzwischen mehrere Male bei Ärzten und Kliniken im ganzen Bundesgebiet installiert. Das Programm ist mittlerweile weiter überarbeitet worden und liegt nun in einer verbesserten Version vor. Über die reine Privatrechnungsschreibung hinaus ist jetzt möglich:

- mit Hilfe der gespeicherten Stammdaten-Karteikarten alle gängigen Formulare zu beschriften
- alle für die Leistungsabrechnung wichtigen Daten für die Kassenpatienten zu erfassen
- Krankenscheine mit den Abrechnungsdaten zu versehen
- jederzeit eine Ziffernstatistik zur Vermeidung von Kassenregressen aufzurufen

An einer Schnittstelle zu 1st_Word wird im Moment noch gearbeitet. Ebenso ist eine intelligente Arzneimittelverwaltung zur Erleichterung der Rezeptschreibung in Planung. Privatliquidation für Ärzte kostet DM 998,–.

Kray-Electronic
Bonhoefferweg 3
3400 Göttingen
Tel.: 05 51 / 2 27 37

ST Archiv

ST Archiv ist eine Programmverwaltung für den ATARI ST und soll helfen, Übersicht in die eigene Software-Sammlung zu bekommen. Das Arbeiten mit ST Archiv geschieht ausschließlich durch die Maus oder alternativ mit einigen wenigen Tasten. Über eine Eingabemaske erreicht man die verschiedenen Funktionen, die dieses Programm anbietet (wie z. B.: Datei laden, File löschen, Anzeige des freien Diskettenplatzes, Drucken von Listen und Diskettenetiketten, etc.). Der Verkaufspreis beträgt incl. verschiedenen Hilfsprogrammen und Bedienungsanleitung DM 79,–.

Gerd Riemann
Heisterberg 3
2248 Hemmingstedt
Tel.: 04 81 / 6 18 66

Nur Aufgeblasen?

Die neuen MEGA-Modelle von ATARI





Bild 2: Die flache Tastatur

Die Weichen sind gestellt, die neuen Mega-ST-Rechner sind im Lande. Schlicht, im gewohnten Grauton und mit einem völlig veränderten Design. Verbirgt sich hinter dieser neuen Gestaltung nur ein alter Rechner mit einem riesigen Speichervermögen, oder hat Atari tatsächlich weitere „Leckerbissen" eingebaut?

Äußerlich sehr kompakt. Modern im Design. Nur die Maus und der Schwarz-Weiß-Monitor erinnern uns an die „alten" ST-Modelle. Die Professionalität, die die Atari-ST-Rechner immer schon hatten, erreicht nun ein Höchstmaß mit den Mega-STs.

Das Keyboard: Die Schnittstelle zum Menschen

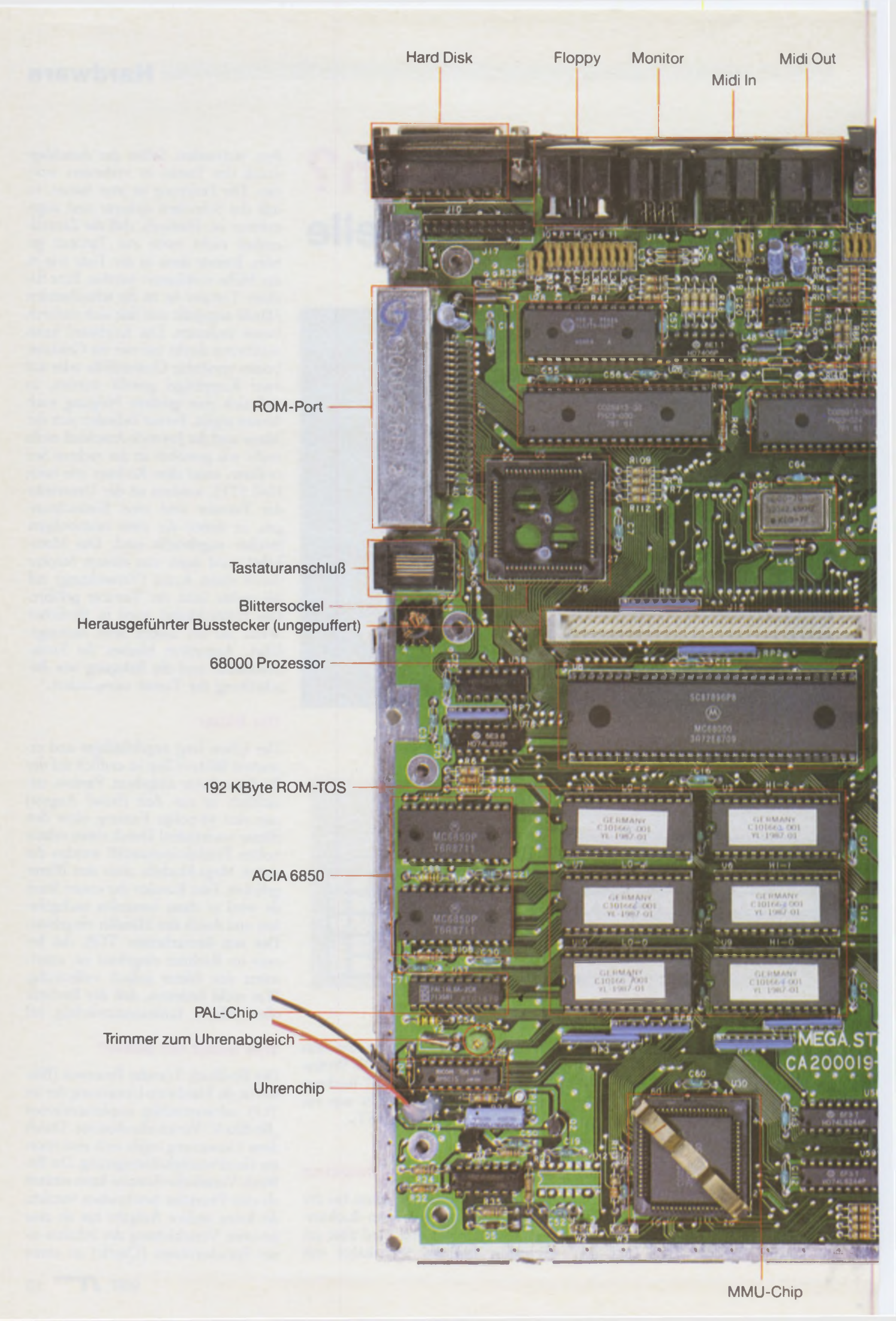
Die Neuerungen fangen schon bei der Tastatur an. Sie wurde vom Rechnergehäuse abgetrennt und wird über ein steckbares flexibles Spiralkabel mit ihm verbunden. Selbst der Anschlagdruck der Tasten ist verbessert worden. Die Federung ist jetzt härter, so daß das Schreiben sicherer und angenehmer ist. Dadurch, daß die Zentraleinheit nicht mehr zur Tastatur gehört, konnte diese in der Tiefe wie in der Höhe verkleinert werden. Eine flachere Tastatur ist an die schreibenden Hände angepaßt und läßt sich dadurch besser bedienen. Das Keyboard kann wahlweise direkt auf vier im Gehäuseboden versenkte Gummifüße oder auf zwei Klappbügel gestellt werden, so daß sich eine größere Neigung nach hinten ergibt. Ferner befinden sich der Maus- und der Joystick-Anschluß nicht mehr wie gewohnt an der rechten Seite (bzw. unter dem Rechner wie beim 1040 STF), sondern an der Unterseite der Tastatur sind zwei Einbuchtungen, in denen die zwei neunpoligen Stecker angebracht sind. Das Mauskabel wird dann von diesem Stecker durch einen Kanal (Versenkung) auf die rechte Seite der Tastatur geführt. Das Joystickkabel wird in ähnlicher Weise auf der linken Seite herausgeführt. Ansonsten blieben die Funktionstasten und die Belegung wie Beschriftung der Tasten unverändert.

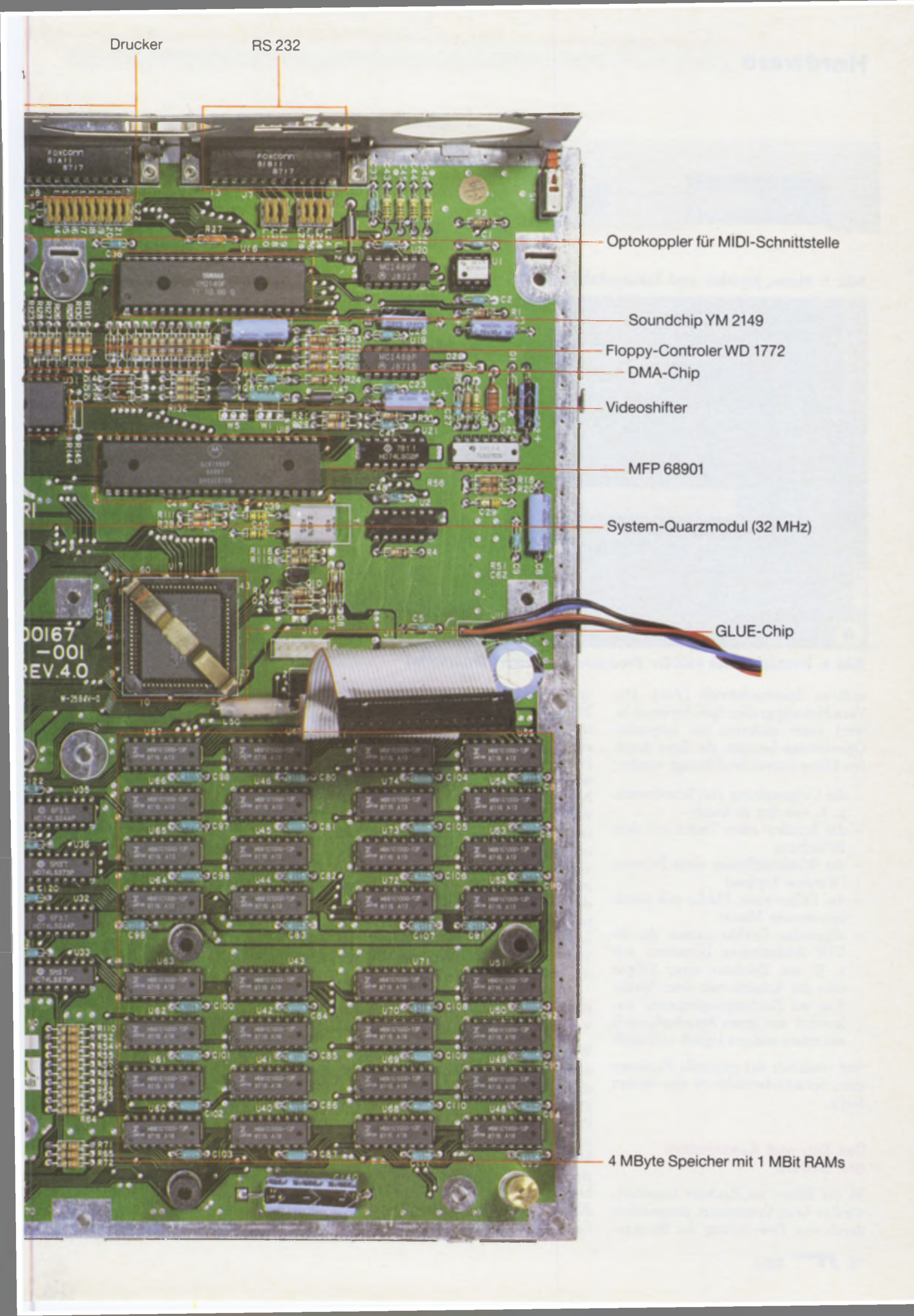
Der Blitter

Der schon lang angekündigte und erwartete Blitter-Chip ist endlich auf der Rechnerplatine eingebaut. Pardon, tatsächlich ist zur Zeit (Stand August) nur eine 68-polige Fassung ohne den Blitter vorhanden! Durch einen relativ hohen Produktionsausfall werden die ersten Mega-Modelle ohne den Blitter geliefert. Den Kunden der ersten Stunde wird er dann kostenlos nachgeliefert und durch den Händler eingebaut. Das neu überarbeitete TOS, das bereits im Rechner eingebaut ist, unterstützt den Blitter jedoch vollständig. Was nicht bedeutet, daß der Rechner ohne Blitter funktionsuntüchtig ist!

Was leistet der Blitter?

Der Bit-Block Transfer Processor (Blitter) ist die Hardware-Umsetzung der im TOS softwaremäßig implementierten „Bit-Block"-Verschiebe-Routine. Durch diese Umsetzung ergibt sich eine enorme Geschwindigkeitssteigerung. Die Bit-Block-Verschiebe-Routine kann einfach als eine Prozedur beschrieben werden, die keine andere Aufgabe hat als eine bitweise Verschiebung des Inhaltes eines Speicherraums (Quelle) an einen


Hard Disk
Floppy
Monitor
Midi In
Midi Out
ROM-Port
Tastaturanschluß
Blittersockel
Herausgeführter Busstecker (ungepuffert)
68000 Prozessor
192 KByte ROM-TOS
ACIA 6850
PAL-Chip
Trimmer zum Uhrenabgleich
Uhrenchip
MMU-Chip
MEGA ST


Drucker
RS 232
Optokoppler für MIDI-Schnittstelle
Soundchip YM 2149
Floppy-Controler WD 1772
DMA-Chip
Videoshifter
MFP 68901
System-Quarzmodul (32 MHz)
GLUE-Chip
4 MByte Speicher mit 1 MBit RAMs

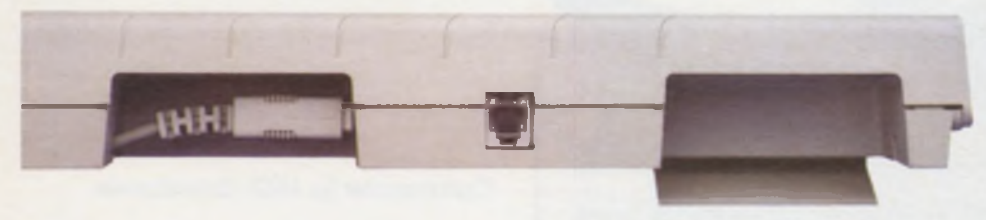


Bild 3: Maus-, Joystick- und Tastaturkabelanschluß

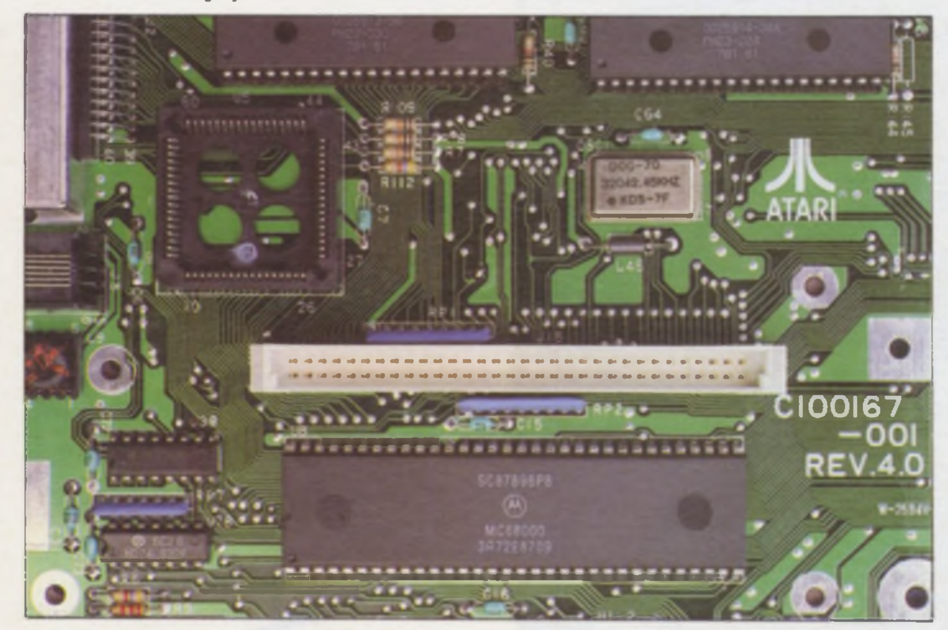


Bild 4: Busstecker mit 68000er Prozessor und leerer Blittersockel

anderen Speicherbereich (Ziel). Die Verschiebung großer Speicherbereiche wird unter anderem bei folgenden Operationen benutzt, die dann durch den Blitter extrem beschleunigt werden:

- die Umgestaltung von Schriftarten, z. B. von fett in kursiv
- das Scrollen eines Textes auf dem Bildschirm
- das Wiederaufbauen eines Fensters (Window Update)
- das Füllen einer Fläche mit einem bestimmten Muster
- allgemeine Grafikroutinen, die die VDI Bibliotheken benutzen, wie z. B. das Zeichnen einer Ellipse oder das Arbeiten mit einer Sprühdose bei Zeichenprogrammen, was letztlich nur einen Speicherbereich mit einem anderen logisch verknüpft

und natürlich das generelle Kopieren eines Speicherbereichs an eine andere Stelle.

Das Ein- und Ausschalten des Blitters

Ist der Blitter im Rechner installiert, wird er beim Systemstart automatisch durch eine Erweiterung des Betriebssystems erkannt. Es erscheint in der Menüleiste unter „EXTRAS" ein weiterer Menüpunkt, mit dem der Blitter ein- und ausgeschaltet werden kann. Ferner läßt sich der Blitter auch softwaremäßig über die XBIOS-Funktion Nr. 64 ($40) ein- und ausschalten (ist Bit0 = 0, so ist der Blitter ausgeschaltet, dementsprechend ist er eingeschaltet, wenn Bit0 = 1 ist). Ist kein Blitter vorhanden, ist jeder Versuch, ihn einzuschalten, sinnlos – es bleibt immer die Softwareroutine (Bitblt) eingeschaltet. Programme, die die VDI- bzw. Line-A Routinen benutzen, werden vom Blitter unterstützt und dadurch im Ablauf beschleunigt.

Nur Zwei- und Vier-Megabyte-Version

Wurde auf der CeBit dieses Jahres noch von einem Mega ST1 gesprochen, so ist dieses Modell nach den jüngsten Aussagen von Atari aus dem Programm gestrichen worden. Als einziger Vertreter der Ein-Megabyte-Klasse bleibt nur der 1040 STF, und als kleinstes Modell der ST-Serie wird weiterhin der 520 STM produziert. Neben diesen beiden Rechnern der „alten" Generation liefert Atari die Mega-Modelle ST2 und ST4. Bei diesen neuen Modellen finden die modernen 1-Megabit-Speicherchips (RAMs) Verwendung. Der einzige Unterschied zwischen den beiden Modellen ist, daß beim ST2 die zweite RAM-Bank unbestückt ist, was bedeutet, daß 16 Speicherchips fehlen.

Genaue Uhrzeit eingebaut

Eine sehr schöne Zugabe bei den Mega-Modellen ist eine bereits eingebaute, quartzgesteuerte Echtzeituhr. Diese Uhr wird durch zwei 1,5 Volt-Batterien oder entsprechende Akkus gepuffert, so daß die Uhr, einmal richtig gestellt, die Zeit und das Datum über Jahre hinaus behält, selbst wenn der Rechner total vom Netz getrennt ist. Akkus werden sogar während des Betriebs vom Rechner nachgeladen. Mit dem gewohnten „Kontrollfeld" wird die Uhrzeit und das Datum einmal gestellt, danach wird die Zeit bei jedem Systemstart vom Betriebssystem gelesen, so daß z. B. bei der Erstellung neuer Dateien immer die aktuelle Zeit und das Datum mit abgespeichert wird.

Weitere Erneuerungen

In Sachen Erneuerungen wurde bei den Mega-Modellen nicht gespart. Die gesammelten Erfahrungen mit den bisherigen ST-Modellen spiegeln sich bei ihrer Konstruktion wider. Um den störenden Kabelsalat verschwinden zu lassen, ist der Rechner samt Floppy mit 720 KByte und integriertem Schaltnetzteil inklusive kleinem Lüfter in ein sehr kompaktes und formschönes Gehäuse, das in seiner Grundfläche kaum größer ist als der Schwarzweißmonitor, untergebracht.
Um auch bei der Elektronik Platz zu sparen und an Effizienz zu gewinnen, sind die Logik-Chips (TTL-Bausteine) der Uhrenansteuerung durch ein einziges sogenanntes PAL ersetzt worden. Weiterhin wurde der Quarzoszillator für den Systemtakt, der vor allem bei den ersten ST-Modellen mit diskreten Bauteilen sehr einfach aufgebaut war und Probleme bereitete (Bildschirmflimmern etc.), durch ein komplett abgeschirmtes Quartzoszillator-Modul ersetzt.
Ebenfalls neu ist der freigelegte Systembus. Neben dem Prozessor befindet sich eine zweireihige Stiftleiste mit insgesamt 64 Anschlüssen, an denen al-



Bild 5: Batteriefach für gepufferte Systemuhr

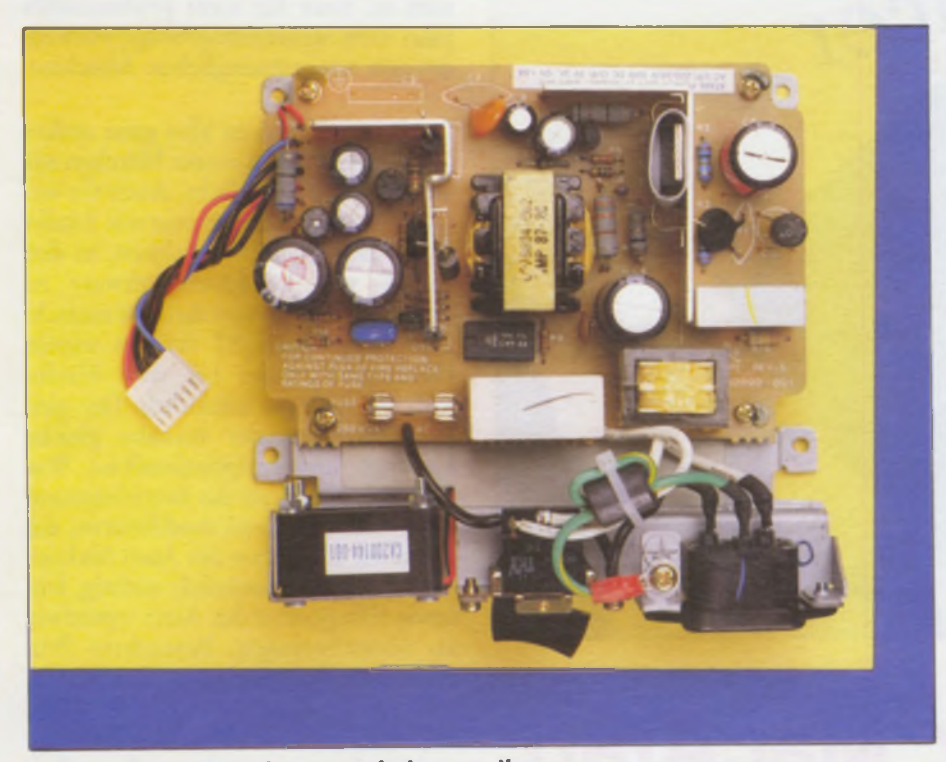


Bild 6: Neues eingebautes Schaltnetzteil

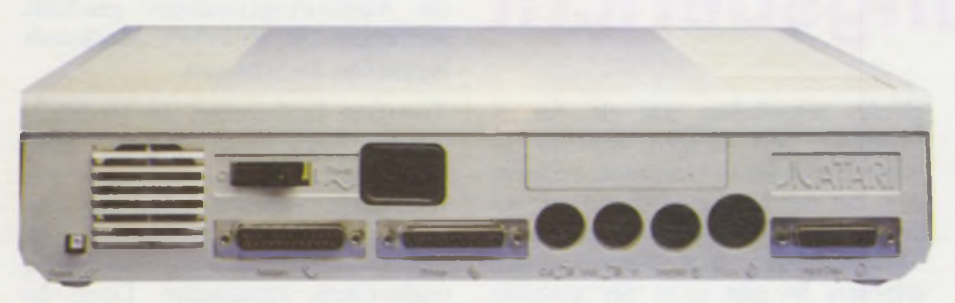


Bild 7: Rechnerrückseite mit allen vorhandenen Schnittstellen und herausnehmbarer Klappe zum Herausführen des Systembus

le Adreß-, Daten- und Steuerleitungen des 68000er Prozessors ungepuffert zur Verfügung stehen. Dadurch ergeben sich mannigfaltige Möglichkeiten, den Rechner zu erweitern bzw. für Meß- und Regelungsprozesse einzusetzen. Die erste Erweiterung, die an diesen Bus angeschlossen wird, kommt von Atari selbst. Es handelt sich dabei um die in der Produktion befindliche „Floating-Point Coprozessor"-Karte. Auf dieser Erweiterung befindet sich der von Motorola hergestellte Coprozessor 68881, der mit einer hohen Frequenz von 20 MHz getaktet wird.

Alles in allem

Atari hat mit den neuen Mega-Modellen nicht nur die Speicherkapazität drastisch erhöht, es handelt sich zudem um eine Erweiterung des bisherigen ST-Konzepts um viele nützlich Features. Damit ist der erheblich höhere Preis dieser Modelle gegenüber dem 1040 STF zu rechtfertigen.

Unserer Meinung nach gibt es ein Manko, das wir doch erwähnen möchten. Im Rechnergehäuse wäre Platz für ein zweites 3,5-Zoll-Floppylaufwerk vorhanden, doch sind weder auf der Frontseite eine Aussparung noch auf der Platine Halterungen für ein solches Laufwerk vorgesehen. Wenigstens die Möglichkeit, eine zusätzliche Floppy einzubauen, hätte das System vervollständigt. So muß der Anwender, der auf eine zweite Floppy nicht verzichten möchte, derzeit auf die bekannte SF 354 bzw. SF 314 mit externem Netzteil zurückgreifen.

Die Preise

Die Mega-Modelle werden nur als komplettes Set mit Monitor SM 124, der Maus und dem neuen ST-BASIC verkauft. Im Rechner ist das neue TOS auf ROMs, der Blitter und eine doppelseitige Floppy mit 720 KByte integriert.

Der Set-Preis für den Mega ST2 beträgt DM 2998 und für den Mega ST4 DM 3998.

(UB/MM)



Aladin, der MacAtari: Ein Erfahrungsbericht

Mit einem kleinen 'Sesam öffne dich' können Sie Ihrem Atari neue Welten erschliessen: Jene Welten, die sonst nach einer bekannten irischen Apfelsorte benannt sind...

Zu einem ersten Test stand uns Aladin schon vor seinem Erscheinungstermin kurze Zeit zur Verfügung. Dennoch wartete die Redaktion, aus purer Neugierde, sehnsüchtig auf ein Serienexemplar. Seit einigen Wochen ist dieses Exemplar in Gebrauch, und seitdem hat der MacIntosh-Emulator aus deutschen Landen auch schon einige Verbesserungen und Veränderungen erfahren. So gibt es in der neuesten Version die Möglichkeit, auch verschiedene andere Drucker als den wenig verbreiteten Apple-ImageWriter zu verwenden. Epsons FX-80, MX-80, LQ-1500 sowie der 24-Nadel-Standard-Drucker NEC P6 werden unterstützt. Bei den 24-Nadlern ist die Qualität sogar um einiges besser als beim Image-Writer.

Die meisten Menschen, die beim Wort 'Maus' nicht auf Stühle steigen, sondern an niedliche Plastikwesen denken, werden schon einmal mit dem MacIntosh von Apple geliebäugelt haben. Die meisten Interessenten werden jedoch nach einem weniger freundlichen Blick auf das eigene Bankkonto auf den Kauf des niedlichen Schuhkartons mit Bildschirm verzichtet haben. Es ist ja nicht nur der Karton, der so teuer ist: Auch die Betriebskosten des Rechners, vor allem wegen horrender Preise für zugegebenermaßen hervorragende Software, machen den MacIntosh zu einer für nicht professionelle (also nicht steuerabzugsfähige) Anwendungen unerschwinglichen Maschine.

Doch inzwischen ist alles ganz anders gekommen, wie in einem Märchen aus Tausendundeiner Nacht. Auch die weniger Betuchten haben nun mit Aladin eine völlig legale Möglichkeit, in den Genuß von MacIntosh-Software zu kommen. Der Grund dafür ist einfach. Der billige Atari ST ist von seinem Hardware-Konzept her dem MacIntosh gar nicht so unähnlich: Der gleiche Prozessor, der beinahe gleiche Speicher- und Peripherieaufbau. Was liegt also näher, als das Betriebssystem des MacIntosh so zu modifizieren, das es auf der Hardware des Atari funktioniert? Dabei ist natürlich wichtig, fehlende Fähigkeiten des Atari irgendwie zu überschminken. Besonderes Problem: Die Tonerzeugung. Der MacIntosh hat zu diesem Zweck vier D/A Wandler, die sich vor denen des Amiga keineswegs verstecken müssen. Leider ist der Atari mit so edlen Teilen nicht ausgestattet. Immerhin haben es die Aladin-Programmierer geschafft, zumindest die einfachen MacIntosh-Signaltöne zu erzeugen.

Trotzdem funktioniert das Modifizieren des Betriebssystems gerade beim MacIntosh sehr gut, weil Apple von vornherein versucht hatte, die Programmierer dazu zu erziehen, auf die MacIntosh-Hardware nur über das Betriebssystem zuzugreifen. Doch der Teufel steckt im Detail. Programme, deren Schöpfer sich brav und genügsam an die Programmierempfehlungen

von Apple gehalten haben, funktionieren völlig problemlos mit dem angepaßten MacIntosh-Betriebssystem. Was passiert aber nun, wenn ein Programm versucht, direkt auf irgendwelche Hardware des MacIntosh zuzugreifen? In diesem Fall versucht die Emulator-Software, die auftretenden Fehler abzufangen und die vom Programm gewünschte Aktion mit Hilfe der Atari-Hardware durchzuführen. Wenn das nicht möglich ist, passiert eben gar nichts; zum Beispiel immer dann, wenn ein Programm auf die MacIntosh-Tonerzeugung zugreift.

Ein weiterer großer Unterschied zwischen Apple und Atari ist das Diskettenformat. Im MacIntosh werden Diskettenlaufwerke verwendet, die mit keinem Standard auf dem Markt kompatibel sind. Da läßt sich auch mit der raffiniertesten Software nichts machen. Wie bekommt man aber nun die MacIntosh-Software in den Atari, wenn der die Disketten nicht lesen kann? Leider führt hier kein Weg an einem 'richtigen' MacIntosh vorbei. Von ihm aus müssen Programme und Daten über die seriellen Schnittstellen der Rechner übertragen werden. Dieser Prozeß dauert ungefähr zwölf Minuten und ist natürlich nur mit nicht kopiergeschützten Programmen möglich.

Aladin besteht aus einer kleinen Platine, die außer zwei leeren IC-Fassungen nur noch ein PAL (vermutlich als Kopierschutz) enthält. Dazu werden drei Disketten mit Software geliefert. Eine enthält den eigentlichen Emulator, eine weitere ist im MacIntosh-Format und enthält das übertragungsprogramm für Disketten über die serielle Schnittstelle, die letzte ist mit einem Demoprogramm bespielt.

Um mit dem Emulator überhaupt etwas anfangen zu können, sind ROMs mit dem originalen MacIntosh-Betriebssystem nötig. Diese werden in die Fassungen auf der Platine, die übrigens am Romport angeschlossen wird, eingesetzt. Doch woher bekommt man nun legal ROMs mit dem Apple-eigenen Betriebssystem? Glücklicherweise hat Apple vor kurzem eine neue Betriebssystem-Version herausgegeben, die Aufrüstung auf den MacIntosh plus nämlich. Aus dieser Umrüstaktion stammen ungeheure Mengen gebrauchter ROMs, man sollte die Chips übrigens auch völlig legal als Ersatzteil kaufen können (aber natürlich nicht einfach kopieren!). Damit haben wir natürlich eine kleine Einschränkung des Aladin verraten, die übrigens auch schon mit die gravierendste ist: Aladin läuft nur mit dem Betriebssystem des alten Mac; die neueste Version, die Software des Mac plus, ist nicht verfügbar. Dies ist im wesentlichen eine Einschränkung in Sachen Geschwindigkeit, es gibt bis heute nur sehr, sehr wenig MacIntosh-Software, die unbedingt einen Mac plus erfordert. Auch mit dem alten MacIntosh läßt es sich gut leben.

Aber jetzt auf zum fröhlichen Emulieren: Emulator in den Romport stecken, Atari anschalten, Diskette mit dem Emulator booten. Auf dem Bildschirm erscheint ein Werbespot, in dem sich ein Atari durch aus einer Wunderlampe entströmende, glitzernde Sternchen in einen MacIntosh verwandelt, und der darauf hinweist, daß der Emulator durch die Spacetaste gestartet und durch die Taste 'C' konfiguriert wird. Sehr hübsch!

Vorsichtshalber also erst einmal konfigurieren. Hier muß zunächst der im Atari verfügbare Hauptspeicher angegeben werden. Die neueste Aladin-Version kann bis zu 2,5 Megabyte Hauptspeicher verwalten. Als nächstes kann man gleich wieder einen beliebig großen Teil dieses Hauptspeichers für eine Ramdisk verbraten, die den schönen Namen SuperDisk trägt. Diese Ramdisk ist tatsächlich super: Sie ist resetfest und, der besondere Clou, nach einem Absturz oder jedem anderen Grund, aus dem der Reset-Knopf gedrückt wurde (spielende Kinder, Hunde, Mauskabel und was es sonst noch so gibt) kann man auch aus dieser Ramdisk booten, vorausgesetzt es befindet sich auf ihr ein Ordner namens System, der beim MacIntosh wichtige Teile des Betriebssystems enthält. Da der MacIntosh immer wieder irgendwelche Dinge von Diskette nachladen muß, macht sich eine große Ramdisk hier durch riesige Geschwindigkeitssteigerung bemerkbar. Nach einem Druck auf den Reset-Knopf des Atari findet man sich tatsächlich nicht wie erwartet auf dem Atari-Desktop, sondern nach wie vor im MacIntosh-Mode – wenn das keine elegante Programmierung ist! Selbst der Reset des Macs wird emuliert. Im weiteren kann man dem Emulator Anzahl und Größe der angeschlossenen Diskettenlaufwerke verraten. Man sollte hier auch

möglichst nichts heimlich tun. Die Verwendung einer Harddisk ist leider noch nicht vorgesehen, soll aber in einem späteren Update ebenfalls möglich werden.

Auch das Tastaturlayout kann verändert werden; alle internationalen MacIntosh-Layouts stehen zur Verfügung. Schließlich ist es noch möglich, den Druckertreiber zu wählen. Die im Moment möglichen Typen sind oben bereits aufgeführt. Auch die Schnittstelle (seriell oder parallel) kann gewählt werden.

Nach der Konfigurierung kann man den Emulator starten. Nach kurzer Ladezeit verlangt er dann eine Diskette mit dem MacIntosh-System. Dabei fällt noch eine weitere Annehmlichkeit im Emulator-Design auf: Am Original gibt's keinen Knopf für den Diskettenauswurf. Man muß dem Mac per Software befehlen, eine Disk herauszurücken. An den Standard-Atari-Laufwerken dagegen kann nichts und niemand den Benutzer daran hindern, zu jeder beliebigen Zeit eine Diskette aus dem Laufwerk zu nehmen. Was nun? Ganz einfach. Nimmt man dem Atari (oder besser dem Aladin) einfach eine Diskette weg, ohne von der Software dazu aufgefordert zu sein, ertönt ein markerschütterndes Pfeifen, jedenfalls dann, wenn man den Monitorlautsprecher nicht abgedreht hat. Dieses Pfeifen hört erst dann wieder auf, wenn man gehorsamerweise die Diskette zurückgibt. Gibt man allerdings per Software den Befehl zum Auswurf, piept es auch, allerdings nicht dauernd, sondern in Intervallen. Damit der Atari überhaupt einen Diskettenwechsel erkennt, dürfen die Disketten nicht schreibgeschützt sein. Um Disketten nicht unbeabsichtigt zu überschreiben, ist ein Software-Schreibschutz implementiert, der als Accessory im MacIntosh-Desktop immer ereichbar ist. Das funktioniert auch sehr gut; schließlich gibt es ja keine TOS-Programme wie beim Atari, sondern konsequenterweise nur Programme, in denen die Menüleiste verfügbar ist.

Woher nimmt man nun aber eine Diskette mit dem Mac-System im Aladin-Format? Hätte man vor dem Starten des Aladin doch nur das Handbuch richtig gelesen... Also Aladin verlassen und das Überspielkabel nehmen, das Sie hoffentlich zuvor gebastelt haben und das die etwas exotischen Stecker des Mac mit den halbwegs normalen des Atari verbinden kann. Glücklicherweise ist die Anschlußbelegung für die Kabel im Aladin-Handbuch angegeben.

Sie nehmen also Ihr Kabel und verbinden die beiden Rechner. Dann kopiert man sich auf dem Mac die zu übertragende Diskette zusammen. Danach braucht man nur noch das Übertragungsprogramm auf dem Mac und dem Atari zu laden und die Diskette zu übertragen. Dabei muß man nur beachten, daß auf die kleinen Atari-Laufwerke keine ganze Macintosh-Diskette paßt, weil jene 400 KByte, diese aber nur 350 KByte fassen. Nach einiger Zeit, die Übertragung dauert ein wenig, weil die serielle Schnittstelle im Atari nicht so schnell wie die des Mac ist, haben Sie eine fast fertige Aladin-Diskette. Leider funktioniert das normale System so noch nicht auf dem Atari. Die System-Dateien müssen erst mit einem speziellen Programm, das aber ebenfalls mitgeliefert wird, angepasst werden. Leider ist dieses Programm nur im MacIntosh-Format dabei. Also müssen Sie, falls Sie dies nicht rechtzeitig gemerkt haben, nochmal Ihren MacIntosh hervorkramen, die System-Dateien auf dem MacIntosh anpassen und dann die Übertragung wiederholen.

Was geschieht bei dieser Anpassung? Das Anpassungsprogramm ändert diejenigen Stellen in den System-Dateien, die irgendetwas von Aladin wollen, was er nicht kann. Die gewünschten Aktionen werden dann simuliert oder ignoriert. Diese Anpassung ist nicht nur bei den System-Dateien notwendig. Auch eine ganze Reihe von Anwendungsprogrammen benötigt eine Anpassung. In der neuesten Version wurde die Anzahl der Anwendungen, die eine Anpassung benötigen, allerdings drastisch reduziert. Aladin wurde noch weiter verbessert, so daß einige Problemfälle jetzt auch ohne jede Anpassung funktionieren. Selbstverständlich gibt es auch nur für eine begrenzte Anzahl von verbreiteten Programmen eine vorbereitete Anpassung. Viele Software läuft allerdings auch ganz ohne Bearbeitung. Eine vollständige Liste gibt's bei Proficomp. Auch wir konnten natürlich nur eine kleine Zahl von Programmen aus dieser Liste testen, aber bei denen, die wir probiert haben, ging es gut.

Die Installation und Anwendung von Aladin ist, wie beschrieben, sehr komfortabel und auch problemlos. Natürlich wäre es einfach toll, wenn man sich das zeitraubende Überspielen von Daten und Software über die serielle Schnittstelle sparen könnte, aber man kann eben nicht alles haben, schon gar nicht zu diesem niedrigen Preis. Aladin ist sein Geld für die Mac-Fans unter den Atari-Benutzern mehr als wert. Wer allerdings Software benutzen will, die auf spezielle Mac-Features angewiesen ist, wird dennoch, wie vermutlich auch die meisten professionellen Anwender (wegen des Services usw.), einen echten Mac vorziehen. Allerdings bietet Proficomp Aladin auch gar nicht als Mac-Ersatz an, sondern vielmehr als Mac-Erweiterung.

Es gibt sogar eine Reihe von Anwendungen, in denen die Kopie dem Original weit überlegen ist: Durch die höhere Bildschirm-Auflösung des Atari passt hier (endlich!) eine DIN A4-Seite in voller Breite auf den Schirm. Die Arbeit mit Textverarbeitung und Desktop-Publishing, aber auch mit Grafikprogrammen wird dadurch viel angenehmer. Und durch das geschickte Software-Design des Mac ist es auch nahezu allen Programmen möglich, diese höhere Bildschirm-Auflösung vollständig zu nutzen! Theoretisch erlaubt der Mac sogar, mehrere Bildschirme gleichzeitig anzusteuern (was natürlich hardwaremäßig nicht ohne weiteres geht) und ein Fenster zur Hälfte auf dem einen, zur anderen auf dem zweiten Monitor darzustellen. Das ist natürlich nicht sehr sinnvoll. Aber interessant wird es, wenn man z. B. bei einem Grafikprogramm sämtliche Dialoge und Befehle auf dem einen Bildschirm beisammen hat, während die gesamte Fläche des zweiten für die Zeichnung zur Verfügung steht. Aber diese 'advanced Features' des Mac stehen hier ja eigentlich nicht zur Debatte. Wie gesagt, die ungefähr 30 Prozent höhere Auflösung des Atari kann von Aladin bzw. dem MacIntosh-Betriebssystem vollständig genutzt werden.

Außerdem kommt hinzu, daß der Atari in der Programmausführung etwas schneller ist als der Mac. Das liegt zum einen an der grundsätzlich etwas höheren Taktgeschwindigkeit, zum anderen daran, daß der MacIntosh im Original nicht mit voller Geschwindigkeit auf seine ROMs zugreifen kann. Im Vergleich mit dem Mac plus, dessen Betriebssystem gegenüber dem alten Mac um einiges verbessert, also

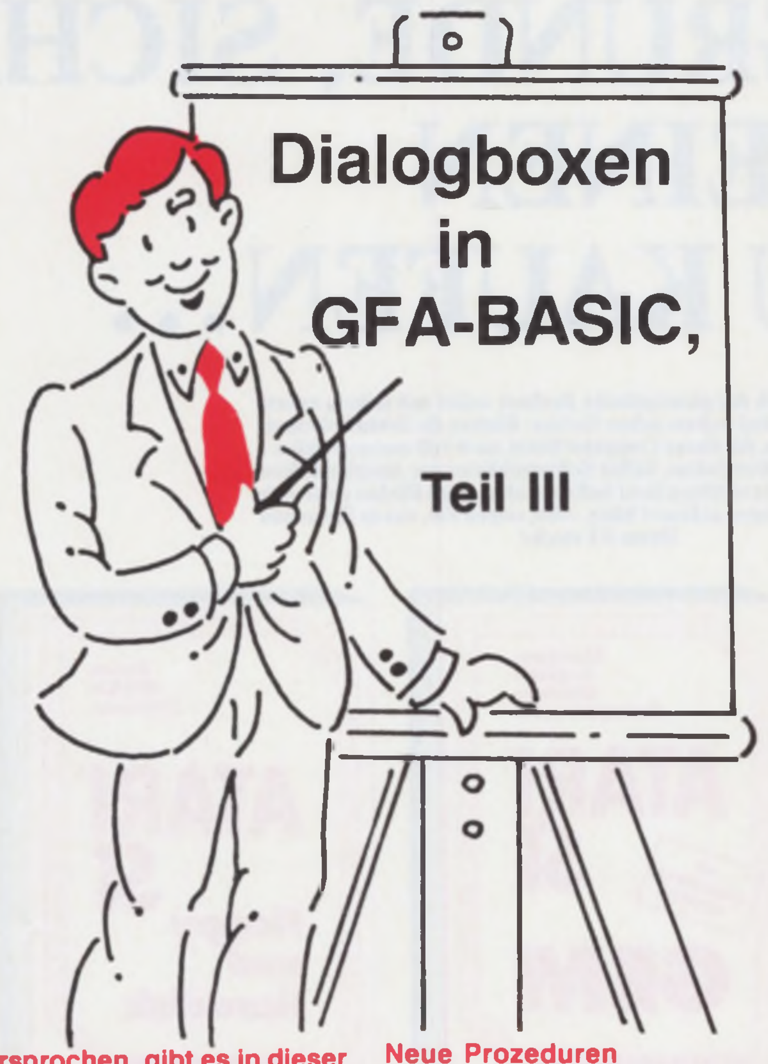
beschleunigt wurde, ist dieser Vorteil aber verschwunden. Vielleicht wird es aber eines Tages mal einen Aladin geben, der auch das neue MacIntosh-Betriebssystem benutzen kann.

Für Programmierer hat die Arbeit auf dem Aladin ebenfalls einen Vorzug: Wer mit Aladin programmiert, muß sich absolut genau an die Apple-Programmierrichtlinien halten, wenn das Programm laufen soll. Jeder unsaubere Zugriff auf das System führt unter Aladin sehr wahrscheinlich zum Absturz. Für die Übertragung eines solchen Programms auf einen richtigen Mac hat dies den unschätzbaren Vorteil, daß dieses Programm mit Sicherheit auf jedem Modell der MacIntosh-Linie läuft, und zwar problemlos – vorausgesetzt, der Speicher des Macs reicht aus (es soll ja auch noch einige 128K-Byte-Uralt-Macs geben). Programmieren mit Aladin geht schon jetzt sehr gut mit der eingebauten Super-Ramdisk und wird noch viel komfortabler werden, wenn erst eine Harddisk angesprochen werden kann.

Intensiv haben wir zum Beispiel Turbo-Pascal getestet; während der Arbeit mit diesem System gab es keinerlei Schwierigkeiten. Es ist wohl anzunehmen, daß auch andere Entwicklungssyteme lauffähig sein werden. Überhaupt ist es erstaunlich, wie viele Programme der Emulator, meist sogar ohne jede Anpassung, verarbeitet. Einzige Voraussetzung ist, daß die Software ohne Kopierschutz vorliegt – kopiergeschützte Software kann mit dem Übertragungsprogramm nicht gelesen und damit auch nicht an den Atari gesendet werden. Das ist deshalb ein Problem, weil auch auf dem Mac immer mehr Programme ausschließlich mit Kopierschutz lauffähig und erhältlich sind. Die übliche Methode, sich dennoch eine Sicherheitskopie zu machen, nämlich die Verwendung eines entsprechend guten Kopierprogramms, scheidet wegen der notwendigen Übertragung über die serielle Schnittstelle ja leider aus.

Auf einem Treffen von Aladin-Fans konnten wir dennoch fast alles, was das Herz begehrt, auf dem Aladin bewundern: Von Microsoft-Produkten wie EXCEL, WORD und CHART über die gängisten Malprogramme MACBILLBOARD, FULLPAINT oder MACDRAW bis hin zu Desktop-Publishing-Programmen wie PAGE MAKER 1.2 oder READY SET GO 3.0 läuft so ziemlich alles nahezu problemlos. Nahezu heißt in diesem Zusammenhang, daß es schon mal ab und zu zur Meldung 'Ein schwerwiegender Systemfehler ist aufgetreten' kommt. Dies geschieht aber selten genug, um mit dem Mac-Emulator sinnvoll arbeiten zu können. Beim Spielen mit ComicWorks haben wir zum Beispiel auch nach anderthalb Wochen noch keinen einzigen Absturz gehabt, genauso mit dem PageMaker. Sehr gut gelöst haben die Aladin-Programmierer auch die Druckeranpassung. Mit den Druckertreibern lassen sich eine Reihe von Standarddruckern sehr gut zusammen mit MacIntosh-Software verwenden. Auch die Druckqualität ist in Ordnung. Was also will man mehr – auch auf der Ausgabeseite fehlt es nicht an Komfort.

(CS)



Dialogboxen in GFA-BASIC, Teil III

Wie versprochen, gibt es in dieser (übrigens vorletzten) Folge wieder neue Anwendungsbeispiele und Prozeduren zur leichten Handhabung von Dialogboxen. Wir wenden uns diesmal der Programmierung eines Schiebereglers und eines neuen Desktops (auch ein Desktop ist im Grunde eine Dialogbox) zu.

Doch zunächst ein Wort in eigener Sache. Für Leser, die neu zu diesem kleinen Kurs hinzukommen, sei unbedingt empfohlen, die beiden ersten Folgen zu studieren und mit den dort vorgestellten Routinen zu experimentieren, da diese ebenso wie der Umgang mit einem Resource-Construction-Set als bekannt vorausgesetzt werden.

Da wir mit Beendigung des letzten Teils im Juli/August-Heft den Löwenanteil der notwendigen Prozeduren bereits hinter uns haben, wollen wir jetzt nur noch spezielle Beispiele mit den eventuell zu ergänzenden Prozeduren vorstellen.

Neue Prozeduren

Graf_slidebox – Verschieben eines Rechtecks in einem anderen Rechteck

Graf_slidebox(Baumadresse%, Außenrechteck%, Schieber%, Richtung%, ★Position%)

Mit Graf_slidebox haben wir wieder einmal eine Original-AES-Routine (Gemsys 76) vor uns. Sie dient dazu, ein Rechteck, also eine Box, innerhalb einer anderen Box zu verschieben, solange eine Maustaste gedrückt bleibt. Hierfür wird eine ganze Reihe von Parametern benötigt. Zunächst wieder einmal die obligatorische Baumadresse im Speicher. Anschließend werden die Objektindizes der beiden Boxen übergeben (oben Außenrechteck% für die Äußere und Schieber für die Innere genannt). Ferner ermöglicht Graf_slidebox eine Bewegung der inneren Box in horizontaler oder vertikaler Richtung. Wir haben einen horizontalen Schieberegler gewählt und mußten deswegen eine Null als Parameter übergeben. Für die vertikale Richtung dient eine Eins. Zurückgegeben wird von der Routine die relative Position des Mittelpunkts unseres Schiebers (X-Position bei horizontaler und Y-Position bei vertikaler Bewegung). Relativ deshalb, weil man einen Wert zwischen 0 und 1000 erhält, den man erst in das richtige Verhältnis setzen muß. Man nimmt dazu die Breite der inneren Box und subtrahiert sie von der der äußeren Box. Anschließend multipliziert man das Ergebnis mit dem Rückgabewert und dividiert das Ganze durch 1000. Daraus ergibt sich dann folgende Formel zur Berechnung:

Ergebnis = (Außenbreite – Innenbreite) ★ Rückgabewert / 1000

Die gleiche Formel wird übrigens zur Berechnung der verschiebbaren Balken bei der Fensterprogrammierung benutzt.

Set_string: Schreibt einen String in eine Dialogbox

Set_string (Baumadresse%, Objektindex%, String$, Stringlänge)

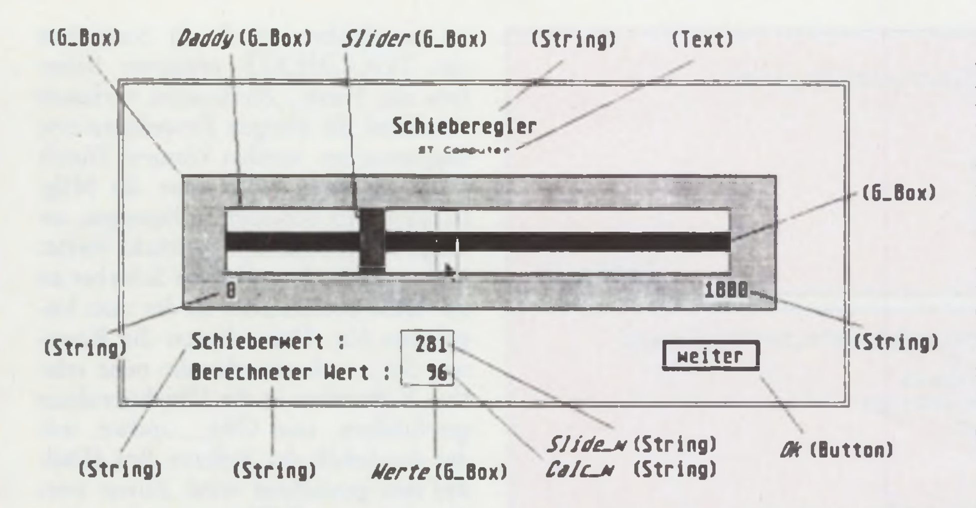
Diese Routine ändert einen in einer Dialogbox definierten String. Benötigt werden zunächst die Baumadresse und der Objektindex. Mit ihnen läßt sich die Adresse des Strings mittels ob_spec (Lpeek auf die Objektadresse plus 12) ermitteln. Der nächste Schritt dient dazu, den String auf die richtige Länge zu bringen. Hierfür sind die Parameter String$ und Stringlänge erforderlich. In einer While...Wend-Schleife wird solange eine Leerstelle vor den eigentlichen String gehängt, bis er die in der Stringlänge definierte Länge hat, im günstigsten Fall also gar keine. Anschließend wird der neue String einfach Zeichen für Zeichen an die errechnete Adresse geschrieben.

Set_xrel: Setzt relative X-Koordinate eines Objektes neu

Set_xrel (Baumadresse%, Objektindex%, X-Koordinate%)

Diese Routine ist die vereinfachte Form einer Prozedur, die wir gleich anschließend vorstellen wollen. Deswegen wird hier nicht näher darauf eingegangen. Will man einen Schieberegler in vertikaler Richtung bewegen, muß man natürlich die relative Y-Koordinate statt der X-Koordinate setzen (Objektadresse% + 18,Yrel%).

Put_objc_xywh: Setzt die relativen X-, Y-Koordinaten, Breite und Höhe eines Objektes



Put__objc__xywh(Baumadresse%, Objektindex%, X%, Y%, Breite%, Höhe%)

Nun also die vollständige Version, die alle Maße eines Objektes bestimmen kann. Die Routine bildet praktisch das Gegenstück zur Prozedur Get__objc__xywh, die wir ja bereits in der letzten Folge kennengelernt haben. In unserem Beispiel benutzen wir sie, um ein neues Desktop zu kreieren bzw. eine Dialogbox auf die richtige Größe zu bringen. Dies ist leider notwendig, da das RCS nicht in der Lage ist, so große Dialogboxen zu erstellen. Die Parameter erfordern wohl keine besondere Erklärung.

Wind__get: Ermittelt Werte von Fensterkomponenten

Wind__get (Windowhandle%, Feld%, ★X%, ★Y%, ★Breite%, ★Höhe%)

Die AES-Routine Wind__get (Gemsys 104) ist recht universell. Sie kommt, wie der Name vermuten läßt, aus der Fensterprogrammierung. Wir wollen uns hier jedoch auf unsere spezielle Anwendung beschränken. Man muß nur soviel wissen, daß die Daten über jedes Fenster in einer sogenannten Rechteckliste vorhanden sind. Jedes Fenster besitzt eine solche Liste. Durch den ersten Parameter Windowhandle% wird bestimmt, welches Fenster angesprochen wird. Wir müssen in unserem Falle eine Null übergeben, da für einen Desktop immer das Windowhandle% Null reserviert ist. Was nun Wind__get aus dieser oben erwähnten Rechteckliste holt, hängt von dem Parameter Feld% ab. Wir übergeben in unserem Beispiel eine Vier, die bewirkt, daß die Koordinaten des gesamten Arbeitsbereiches eines Fenster bzw. des Desktops zurückgegeben werden. Arbeitsbereich bedeutet bei einem Desktop der gesamte Bildschirm bis auf die Menüleiste, die ausgespart bleibt. Will man den ganzen Bildschirm ohne Menüleiste benutzen, was wohl nicht im Sinne von GEM wäre, müßte man als Parameter eine Fünf übergeben. Alle weiteren Möglichkeiten sind für uns momentan uninteressant und würden zu sehr in die Fensterprogrammierung führen.

Wind__set: Setzt Werte von Fensterkomponenten

Wind__set (Windowhandle%, Feld%, Baumadresse%)

Wind__set (Gemsys 105) bildet quasi das Gegenstück zu Wind__get. Normalerweise kann man wieder vier Parameter neben Windowhandle% und Feld% angeben. Das hätte aber bei der Desktop-Programmierung zur Folge, daß die Baumadresse% auf zwei Parameter verteilt würde, also in High- und Low-Word aufgespalten werden müßte. Man hätte sich natürlich auch für diesen Fall eine Routine schreiben können, aber da wir hier keine Fenster programmieren wollen, haben wir darauf verzichtet. Man muß nur darauf achten, diese Routine gegebenenfalls zu modifizieren, damit sie dann universell ist.

Zu den Parametern ist zu sagen, daß wieder das Windowhandle% (also beim Desktop eine Null) zu übergeben ist. Der Feld%-Parameter erhält den Wert 14, was dem AES mitteilt, daß ein neuer Desktop angemeldet wird. Anschließend muß nur noch die Baumadresse des neuen Desktops angegeben werden.

```
Procedure Graf_slidebox(Gr_sltree%,Gr_slparent%,Gr_slobj%,Gr_slvh%,Aes_return)
  '
  Dpoke Gintin,Gr_slparent%
  Dpoke Gintin+2,Gr_slobj%
  Dpoke Gintin+4,Gr_slvh%
  Lpoke Addrin,Gr_sltree%
  Gemsys 76
  *Aes_return=Dpeek(Gintout)
Return


Procedure Set_xrel(Tree%,Obj_index%,Xrel%)
  '
  Obj_adresse%=Tree%+24*Obj_index%
  Dpoke Obj_adresse%+16,Xrel%
Return
```

Listing 3:

```
Procedure Put_objc_xywh(Tree%,Obj_index,X,Y,Width,Height)
  '
  Obj_adresse%=Tree%+24*Obj_index
  Dpoke Obj_adresse%+16,X
  Dpoke Obj_adresse%+18,Y
  Dpoke Obj_adresse%+20,Width
  Dpoke Obj_adresse%+22,Height
Return
```

Listing 4:

```
Procedure Wind_get(Handle%,Field%,Wi_gw1%,Wi_gw2%,Wi_gw3%,Wi_gw4%)
  '
  Dpoke Gintin,Handle%
  Dpoke Gintin+2,Field%
  Gemsys 104
  *Wi_gw1%=Dpeek(Gintout+2)
  *Wi_gw2%=Dpeek(Gintout+4)
  *Wi_gw3%=Dpeek(Gintout+6)
  *Wi_gw4%=Dpeek(Gintout+8)
  Fehler%=Dpeek(Gintout)
Return


Procedure Put_obj_xywh(Tree%,Obj_index,X,Y,Width,Height)
  '
  Obj_adresse%=Tree%+24*Obj_index
  Dpoke Obj_adresse%+16,X
  Dpoke Obj_adresse%+18,Y
  Dpoke Obj_adresse%+20,Width
  Dpoke Obj_adresse%+22,Height
Return
```

Listing 5:

```
Procedure Wind_set(Handle%,Field%,Wi_swl%)
  '
  Dpoke Gintin,Handle%
  Dpoke Gintin+2,Field%
  Lpoke Gintin+4,Wi_swl%
  Gemsys 105
  Fehler%=Dpeek(Gintout)
Return
```

Listing 6:

```
Procedure Set_string(Tree%,Obj_index%,Str_txt$,Laenge%)
  '
  Obj_adresse%=Tree%+24*Obj_index%
  Str_adresse%=Lpeek(Obj_adresse%+12)
  While Len(Str_txt$)<>Laenge%
    Str_txt$=" "+Str_txt$
  Wend
  For I%=0 To Laenge%-1
    Poke Str_adresse%+I%,Asc(Mid$(Str_txt$,I%+1,1))
  Next I%
Return
```

Listing 7:

Ein Schieberegler im GEM-Gewand

Das eigentlich Schwierige beim „Bauen“ eines Schiebereglers ist das Gewußt-wie. Wenn man erst einmal die nötigen „Bauelemente“ hat, ist es eine reine Konstruktionssache mit dem RCS. Auf Bild 1 erkennen Sie, daß der reine Schieber nur aus Elementen des Typs G_box besteht, die ineinander verschachtelt werden. Zur besseren Unterscheidung der einzelnen Boxen haben wir verschiedene Füllmuster ausgewählt. Es ist auch darauf zu achten, das die äußere Box des Schiebers (Daddy) das übergeordnete Objekt zur inneren Box (Slider) ist.

Wichtig ist dabei, daß der eigentliche Schieber, also die innere Box, im Bild Slider genannt, bei der Erstellung als TOUCHEXIT (das TOUCHEXIT-Flag setzen) erklärt wird. Würden wir dies unterlassen, könnten wir unsere Dialogbox nur über den Weiter-Button verlassen und unser Schieber ließe sich nicht eine Spur bewegen.

Wegen wiederholter Anfragen noch einmal kurz zum Erstellen der Dialogboxen und zu den Bildern: Die Objekte der Dialogbox sind in Klammern geschrieben. Ist es notwendig, daß ein Objekt mit einem Namen versehen wird, steht dieser Name kursiv vor dem Objekt. Soweit zum Format der Bilder.

Benutzt werden im ersten Programmbeispiel folgende Routinen:
Rsrc_load, Rsrc_gaddr, Box_draw, Form_do, Form_center, Form_dial, Objc_draw, Rsrc_free, Box_undraw, Get_objc_xywh, Objc_update, Objc_offset, Graf_slidebox, Set_string, und Set_xrel.

Obligatorisch sind wie immer die Prozeduren Rsrc_load, Rsrc_gaddr und Box_draw, so daß diese hier nicht mehr weiter besprochen werden sollen. Doch dann wird es schon interessanter. Als erstes benötigen wir die Breiten des Schiebers und der Box um den Schieber, um damit später mit der oben angegebenen Formel den Rückgabewert von Graf_slidebox richtig zu berechnen. Dies erreichen wir durch die Aufrufe der Prozedur Get_objc_xywh. Das Ergebnis wird der Variablen Pos_alt zugewiesen. Ferner wird als Bewegungsrichtung horizontal ausgewählt.

Nun kommen wir zur Do...Loop-Schleife. Sie wird erst dann verlassen, wenn der weiter-Button (Ok) angeklickt wird. Klickt man also auf den Schieber der Box, bleibt man immer noch in der Schleife. Dadurch wird es uns ermöglicht, den Regler überhaupt zu verschieben, da durch Anklicken des TOUCHEXIT erklärten Schiebers die Form_do-Routine verlassen wird und die übrigen Prozeduren erst angesprochen werden können. Durch Graf_slidebox erhält man die Möglichkeit, den Schieber zu bewegen, solange ein Mausknopf gedrückt bleibt. Läßt man ihn los, wird der Schieber an die Stelle positioniert, an der man losgelassen hat. Dazu dienen die Routinen Set_xrel, mit der die neue relative X-Position in die Objektstruktur geschrieben, und Objc_update, mit der der Inhalt der äußeren Box (Daddy) neu gezeichnet wird. Zuvor werden der neue X-Wert nach der oben erwähnten Formel berechnet, der Original-Rückgabewert und der berechnete Wert mittels Set_string in die Box geschrieben und die Box um diese beiden Strings neu gezeichnet. Am Ende kommen dann wieder die gewohnten Routinen Box_undraw, Undo_objc und Rsrc_free.

Ein eigener Desktop

Viele mögen sich hier vielleicht fragen, wozu ein eigener Desktop überhaupt notwendig sei. Er bringt einen wesentlichen Vorteil mit sich: Der sogenannte Screenmanager (Bildschirmverwalter) nimmt unseren neuen Desktop unter seine Fittiche. Das heißt zum Beispiel, daß alle Objekte, die in unserem Desktop integriert sind, automatisch neu gezeichnet werden, falls sie durch irgendeine Box oder ein Fenster überlagert waren. Das dürfte im häufigsten Falle bei Icons (z. B. Laufwerk-Icons) oder Funktionstasten la 1st_Word auftreten. Kurz gesagt, wir brauchen uns nicht mehr um das Retten des



ATARI ST Bibliothek

Ob frischgebackener ST-Besitzer oder ambitionierter 68000er-Programmierer – wer seinen ATARI ST effizient und professionell einsetzen will, der braucht hochrangige Informationen von hochkarätigen Autoren. Informationen, die genau auf seine Bedürfnisse und Anwendungsgebiete zugeschnitten sind. Informationen, die Sie in unserer ST-Bibliothek finden können. Eine kleine Auswahl daraus stellen wir Ihnen hier vor.



Das erste Buch zum ATARI ST ist für viele das wichtigste. Denn nur ein richtiger Einstieg garantiert später die volle Nutzung dieses Superrechners. ATARI ST für Einsteiger ist eine leichtverständliche Einführung in Handhabung, Einsatz und Programmierung Ihres ST: Von der Tastatur und der Maus hin zum Desktop bis zum ersten BASIC-Befehl und schließlich zum ganzen Programm. Die Autoren zeigen Ihnen den Weg zum Erfolg mit Ihrem neuen Rechner.
ATARI ST für Einsteiger
248 Seiten, DM 29,–



Tips und Tricks zum ATARI ST – das Buch, das voller guter Ideen steckt. Sie suchen einen Druckerspooler? In diesem Buch finden Sie ihn. Sie brauchen eine Farb-Hardcopy-Routine? Bitteschön. Automatisches Starten von TOS-Anwendungen? Nichts leichter als das. Eine RAM-Disk anlegen? Nur zu. Viele neue Routinen, die problemlos in eigene Programme eingebunden werden können, warten nur darauf, daß Sie sie ausprobieren.
ATARI ST Tips & Tricks
Hardcover, 352 Seiten, DM 49,–



68000-Assembler – der Schlüssel zur Programmierung der spektakulärsten Effekte und der leistungsfähigsten Programme auf dem ATARI ST. Dieses Buch zeigt Ihnen, wie Sie Ihren ST mit Maschinensprache voll ausnutzen können: Verwendung von Systemroutinen, Bitmanipulationen, Rekursion, Stacks, Einbinden von Assemblerroutinen in Hochsprachen – vom ersten Schritt bis hin zum echten Einsatz von Maschinensprache: dieses Buch begleitet Sie.
Das Maschinensprachebuch zum ATARI ST
334 Seiten, DM 39,–



CAD ist mehr als „nur" Computergrafik. Neben den speziellen Programmiertechniken und den typischen CAD-Prozeduren braucht man noch solide Grundkenntnisse zum Aufbau eines CAD-Systems. Wissen, das in diesem Buch kompakt und leichtverständlich zusammengefaßt wurde. Zudem können Sie anhand der einzelnen beschriebenen Module leicht ein komplettes CAD-Programm erstellen. So wird das Gelernte gleich in die Praxis umgesetzt. Rationelles Arbeiten mit CAD? Mit diesem Buch kein Problem.
ATARI ST – Einführung in CAD
Hardcover, 289 Seiten
inkl. Diskette, DM 69,–



Der DATA BECKER Führer zum ATARI ST,
240 Seiten, DM 29,80



Der DATA BECKER Führer zu 1st Word,
192 Seiten, DM 24,80

Bestimmt kennen Sie das Problem: Sie sind mitten beim Programmieren und suchen eine ganz bestimmte Information. Und dann geht die Sucherei los. Eigentlich kann es nur im Handbuch gestanden haben. Oder war es in einer Zeitschrift? Wenn Sie ganz gezielt Informationen suchen, dann haben wir die richtige Quelle für Sie. Die DATA BECKER Führer.



Die Grafikfähigkeit des ST gezielt für eigene Anwendungen einsetzen – dieses Buch zeigt Ihnen, wie es geht. Angefangen von den Grundlagen des VDI, GEM, AES und TOS bis hin zu speziellen Problemlösungen wie Programmierung des Rasterinterrupts oder einer flackerfreien Animation finden Sie hier alles zum Thema Grafik auf dem ST. Mit zahlreichen Utilities in GFA-BASIC, C und Assembler.
Das Supergrafikbuch zum ATARI ST
Hardcover
838 Seiten
inkl. Diskette
DM 69,–

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name
Straße
Ort

ST 97

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

```
' ********************************
' *                              *
' *   Dialogboxen in GFA-BASIC   *
' *                              *
' *   Teil III  (Schieberegler)  *
' *                           HE *
' *                              *
' ********************************
'
Baum=0      !    TREE
Daddy=2     !    OBJECT in TREE #0
Slider=4    !    OBJECT in TREE #0
Ok=9        !    OBJECT in TREE #0
Werte=12    !    OBJECT in TREE #0
Slide_w=13  !    OBJECT in TREE #0
Calc_w=14   !    OBJECT in TREE #0
'
Gosub Rsrc_load("sept871.rsc",*Fehler%)
If Fehler%=0 Then
  Print "RSC-Ladefehler"
  End
Endif
'
Gtype%=0
Gindex%=Baum
Gosub Rsrc_gaddr(Gtype%,Gindex%,*Baum_adr)
'
Rette=1
Gosub Box_draw(Baum_adr,0,0,0,0)
Rette=0
Gosub Get_objc_xywh(Baum_adr,Slider,*X,*Y,*W,*H)
Gosub Get_objc_xywh(Baum_adr,Daddy,*Xp,*Yp,*Wp,*Hp)
Hor=0 ! horizontale Bewegung
Pos_alt=Wp-W
Do
  Exit If Ex_obj%=Ok
  Gosub Form_do(Baum_adr,0,*Ex_obj%)
  Gosub Graf_slidebox(Baum_adr,Daddy,Slider,Hor,*Pos)
  Pos$=Str$(Pos)
  Gosub Set_string(Baum_adr,Slide_w,Pos$,4)
  Pos_calc=(Pos_alt)*Pos/1000
  Pos_calc$=Str$(Int(Pos_calc))
  Gosub Set_string(Baum_adr,Calc_w,Pos_calc$,4)
  Gosub Objc_update(Baum_adr,Werte)
  Gosub Set_xrel(Baum_adr,Slider,Pos_calc)
  Gosub Objc_update(Baum_adr,Daddy)
Loop
Gosub Box_undraw(Baum_adr,0,0,0,0)
Gosub Undo_objc(Baum_adr,Ex_obj%,1)
'
Gosub Rsrc_free
End
```

Listing 1:

```
' ********************************
' *                              *
' *   Dialogboxen in GFA-BASIC   *
' *                              *
' *   Teil III  Eigener Desktop  *
' *                           HE *
' *                              *
' ********************************
'
Desk=0  !   TREE
Gosub Rsrc_load("sept872.rsc",*Fehler%)
If Fehler%=0 Then
  Print "RSC-Ladefehler"
  End
Endif
Gtype%=0
Gindex%=Desk
Gosub Rsrc_gaddr(Gtype%,Gindex%,*Baum)
Ob_width=640
Ob_height=400
Obj_index=0
Gosub Put_objc_xywh(Baum,Obj_index,0,0,Ob_width,Ob_height)
Gosub Wind_get(0,4,*Xdesk%,*Ydesk%,*Wdesk%,*Hdesk%)
Gosub Wind_set(0,14,Baum)
Gosub Objc_draw(Baum,0,8,Xdesk%,Ydesk%,Wdesk%,Hdesk%)
Alert 1,"Dies ist ein eigener Desktop! |    in GFA-BASIC !!!",1,"weiter",A
Gosub Rsrc_free
End
```

Listing 2:

Hintergrunds zu kümmern. Aber dies trifft nur bei Objekten zu, die in unserem Desktop integriert sind, die also schon in unserer Dialogbox vorhanden sind, wenn sie als Desktop angemeldet wird, nicht bei nachträglich auf den Desktop gezeichneten Objekten. Wir haben als Beispiel ein sogenanntes Image in den Desktop eingebaut, das zuvor mit einem Icon-Editor erstellt wurde. Doch nun zur Programmierung.

Benutzt werden im zweiten Programmbeispiel folgende Routinen: Rsrc_load, Rsrc_gaddr,Objc_draw, Rsrc_free, Put_objc_xywh, Wind_set und Wind_get.

Nach Rsrc_load und Rsrc_gaddr wird zunächst die Größe des Bildschirms bestimmt, hier für einen monochromen Bildschirm 600x400 Pixel. Will man einen Farbschirm benutzen, müssen hier andere Werte entsprechend der Auflösung eingetragen werden, oder man fragt die Auflösung mittels Xbios(4) ab und setzt die entsprechenden Werte dann ein. Mit Put_objc_xywh werden die Werte in die Objektstruktur geschrieben. Die Koordinaten des Arbeitsbereiches für unseren Desktop holen wir uns anschließend über Wind_get. Die Prozedur meldet dann den neuen Desktop an und mit Objc_draw wird er auf den Bildschirm gebracht. Jetzt kann irgendein Programm folgen, bevor mit Rsrc_free der Speicher wieder freigegeben wird.

Somit genug für diesmal. Im Oktoberheft werden wir uns, im letzten Teil dieser Reihe, mit zwei weiteren Beispielen beschäftigen, die aber jetzt noch nicht verraten werden. Bis dann!

(HE)

EUMEL-Betriebssystem für den Atari ST

Teil 2: Installation

Nachdem in der vergangenen Ausgabe das neue Multitasking-Betriebssystem EUMEL theoretisch vorgestellt wurde, soll dem Trockenschwimmen nun der Sprung ins kalte Wasser folgen. Unser Testexemplar kam in einem großen Paket: Drei Schuber mit jeweils rund 300 Seiten hervorragend aufbereiteter Anleitung, ein ROM-Modul und 6 Disketten. Die gelieferte Software bestand aus

- dem eigentlichen Betriebssystem ELAN (Hintergrund und Urlader),
- dem Editor mit allen Paketen zur Textkosmetik und 20 Druckertreibern sowie Spool-Software für die Druckertask,
- dem Taschenrechner TeCal für den Editor,
- dem Datenauswertungssystem DASYS,
- der Modellsprache DYNAMO,
- dem MS-DOS Transferpaket,
- der KI-Sprache LISP und
- dem HAMSTER, einer Roboteremulation.

Wir testeten die EUMEL-Version 1.8/M. Das „M" steht für Multiuser und bedeutet, daß eine Festplatte unbedingt erforderlich ist. An dieser Stelle bedanke ich mich herzlich bei der Firma CSF in Bielefeld, die mir freundlicherweise eine Festplatte zur Verfügung stellte, da meine defekt war. Auch ein System im „Mini-Mode" ist lieferbar, auf den Festplatteneinsatz kann dann – zu Lasten der Arbeitsgeschwindigkeit – verzichtet werden.

Die Installation von EUMEL verlief insgesamt recht problemlos. Zunächst wurde die zweite Partition der Harddisk (Laufwerk D) für das neue Betriebssystem reserviert, d. h. das Desktop-Icon gelöscht. Dieser Schritt ist unbedingt notwendig, um eine (absturzträchtige) Kollision von GEM und EUMEL zu verhindern. Aus dem ROM-Modul konnte EUMEL mit einem Klick gestartet werden. Dann fordert ein Installationsmenü den sogenannten EUMEL-„Hintergrund" und den „Urlader" von zwei Disketten an, und endlich meldet sich das Betriebssystem mit der typischen „EUMEL-Tapete", die auf Bild 1 wiedergegeben ist. Nur am Rande bemerkt: Die Bildschirmhardcopies haben mir zwei schlaflose Nächte bereitet, da EUMEL die VBL-Queue kappt, die das DUMP-Programm von Peter Melzer benötigt. Daß es dennoch geklappt hat, verdanke ich einem Tip von Oliver Schildmann – merci beaucoup!

Damit ist die Installationsphase keinesfalls beendet. Da EUMEL – wie in der letzten Folge bereits erwähnt – mit fast jedem Terminal zusammenarbeitet, muß man das EUMEL-System bei der Inbetriebnahme konfigurieren. Aus den 30 (!) mitgelieferten Terminalvoreinstellungen wählten wir für den Atari ST den „psi"-Modus für Kanal 1 (Bildschirm). Zwei kleine Programme waren erforderlich, um EUMEL beizubringen, daß der ST-Rechner über einen 25-Zeilen-Bildschirm verfügt. An der seriellen Schnittstelle (Kanal 2) ist ein weiteres Terminal im Multiuser-Betrieb anschließbar; prinzipiell sollte es mit einer Erweiterungsbox möglich sein, bis zu zehn Terminals an der EUMEL-Atari-Station zu betreiben.

Als Supervisor-Tasks wurden Archiv und Drucker (Kanal 15) eingerichtet. Für letzteren sieht man in der Regel eine Spool-Task vor, die alle Benutzeraufträge sammelt (der Anwender kann in seiner Task sofort weiterarbeiten) und in der Reihenfolge des Eintreffens zum Drucker schickt. Die eigentliche Druckeranpassung unter EUMEL ist erheblich komplizierter als bei herkömmlichen Textprogrammen; für den NEC P 6 war ein ELAN-Programm mit 550 Zeilen erforderlich (ausschnittsweise in Bild 2 dargestellt). Glücklicherweise findet man auf den Disketten fertige Treiber, u. a. für alle EPSON-kompatiblen Geräte, den Apple-Imagewriter und auch für zwei Laserdrucker: Den HP-Laserjet (plus) und den Kyocera F1010.

Soweit die wesentlichen Installationsschritte. Bei Bedarf kann jeder Benutzer zusätzliche Funktionen implementieren, auf die später teilweise eingegangen wird (Vernetzung, MS-DOS-Dateitransfer). Bis zur Inbetriebnahme der EUMEL-Station brauchte ich ungefähr drei Tage. Zwar sind in den Handbüchern alle Prozeduren detailliert erklärt, aber wer sich zum erstenmal mit EUMEL beschäftigt, sollte schon vorher wissen, daß eine gründliche theoretische Einarbeitung in das

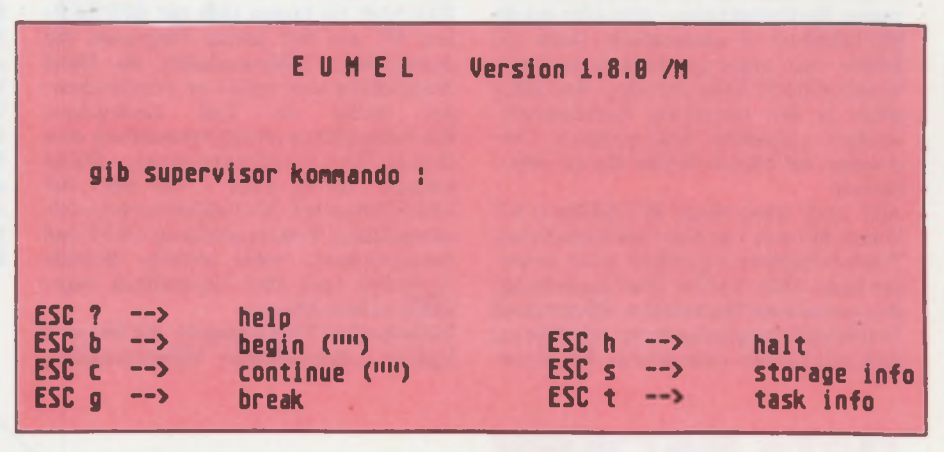


Bild 1: Die „EUMEL-Tapete" am Anfang einer Arbeitssitzung



Weil Sie
Ihren ATARI ST
zu 100 % nutzen wollen
SYBEX
GFA-BASIC
Referenz-Handbuch
Michael Koller
Michael Koller
GFA-BASIC Referenz-Handbuch
NEU
Wenn Sie alles aus GFA-BASIC herausholen wollen, brauchen Sie keine verschiedenen Quellen: In diesem umfangreichen Arbeits- und Nachschlagewerk finden Sie wirklich alle zum Programmieren notwendigen Informationen gebündelt. Dabei ist die logisch geordnete Befehlsliste nur ein Bestandteil des Buches; zusätzlich werden schwer bedienbare Befehle und Funktionen sehr ausführlich beschrieben, wobei der Autor sinnvolle Beispiele hinzugefügt hat. Außerdem gibt er zusätzliche Informationen über Programmiertechniken; damit Probleme bei der GEM-Programmierung (die auch beschrieben werden) Ihre Arbeit nicht unnötig verzögern, bietet Michael Koller Ihnen direkt die zur Lösung erforderlichen Kenntnisse über das Betriebssystem. Highlights sind u. a. die Verwendung der RSC-Dateien und eine Fensterverwaltung mit Beispielprogramm. Erfahren Sie, was Ihr ATARI ST mit GFA-BASIC tatsächlich leisten kann!
528 Seiten / zahlr. Abb., Best.-Nr. 3555
ISBN 3-88745-555-X (1987), DM 49,– / sFr. 45,10 / S 382,–
Übrigens:
SYBEX sucht ständig gute Buch- und Software-Autoren.
Interessiert?
Dann kontakten Sie bitte Ralf Lieder,
Tel. 0211/618 02 20.
SYBEX
inkl. PowerPoint
Programmentwicklung mit
GFA-BASIC
Klaus Löffelmann
Klaus Löffelmann
Programmentwicklung mit GFA-BASIC
inklusive Grafikprogramm PowerPoint
NEU
Wenn Sie sich in die professionelle Programmierung unter GFA-BASIC ganz praktisch einführen lassen wollen – Klaus Löffelmann hat für Sie ein Anwendungs-Paket mit Mehrfachnutzen geschaffen: Anhand eines sehr komfortablen Grafikprogramms (PowerPaint) weiht er Sie schrittweise in die Geheimnisse der Programm-Entwicklung mit GFA-BASIC unter der Benutzer-Oberfläche GEM ein. Dabei ist nicht nur das vollständige Listing des Programms ein zentraler Buchteil – PowerPaint wird auf der integrierten Diskette direkt mitgeliefert. Natürlich hat der Autor neben dieser handfesten Praxisunterweisung weder eine leicht verständliche Einführung in GFA-BASIC noch eine ausführliche Befehlsreferenz vergessen – teilweise mit umfangreichen Beispielprogrammen, die ebenfalls auf der Diskette enthalten sind. Lernen – entwickeln – anwenden: So erschließen Sie sich die Möglichkeiten von GFA-BASIC richtig.
ca. 350 Seiten + Programm-Diskette, Best.-Nr. 3510
ISBN 3-88745-510-X (1987), DM 58,– / sFr. 53,40 / S 452,–
SYBEX
GEM
Programmier-Handbuch
Phillip Balma
William Fitler
Phillip Balma / William Fitler
GEM Programmier-Handbuch
NEU
Ein unentbehrliches Handbuch für im Umgang mit C vertraute Leser, die genaue Informationen über die grafische Benutzer-Oberfläche GEM benötigen. Angesprochen sind damit alle Nutzer von Systemen, auf denen GEM implementiert ist: dazu gehören der ATARI ST, IBM PC sowie der starke Schneider PC – und zu diesen kompatible Rechner. Nützliche Programm-Beispiele dokumentieren die wichtigen Lernschritte. Nach der Lektüre können Sie nachvollziehen, wie Sie Objekte, Fenster, Dialogboxen und andere Utilities aufrufen und in eigene Programme einbauen. Mit einer Übersicht über die VDI- und AES-Routinen, einer Einführung in den Gebrauch des Resource Construction Set sowie dem Source Code für das im Buch enthaltene Zeichenprogramm.
In Kürze
ca. 550 Seiten / 54 Abb., Best.-Nr. 3692
ISBN 3-88745-692-0 (1987), DM 59,– / sFr. 54,30 / S 460,–
SYBEX
ATARI ST
Programmieren in Maschinensprache
Christian Nieber
Christian Nieber
ATARI ST
Programmieren in Maschinensprache
NEU
Erschließen Sie sich mit diesem Buch die Programmierung in Maschinensprache auf Ihrem ATARI ST. Der Autor führt in leicht verständlicher Art in die Entwicklung von Maschinensprache-Routinen für die ST-Systeme ein. Er erläutert den Aufbau und die Funktionsweise des Mikroprozessors MC 68000 sowie dessen Befehlssatz und Adressierung. Weitere Schwerpunkte sind die Programmierung von Grafik & Sound und die GEMDOS-, BIOS- und XBIOS-Routinen. Alle wichtigen Funktionen werden anhand von kleinen, in Assembler entworfenen Routinen erläutert, die nach der Einbindung in die Benutzeroberfläche des ATARI ST lauffähig sind.
432 Seiten / mit Abb., Best.-Nr. 3678
ISBN 3-88745-678-5 (1987), DM 48,– / sFr. 44,20 / S 374,–
Weitere SYBEX-Bücher für Ihren ATARI ST
3623 Arbeiten mit dem ATARI ST DM 49,–
3676 ATARI ST – Einführung WordStar DM 48,–
3665 ATARI ST – Arbeiten mit CP/M DM 48,–
3642 Das Floppy-Arbeitsbuch (+ Diskette) DM 69,–
3674 ATARI ST – Tips & Tools zu C DM 48,–
3626 GEM: Die AES-Bibliothek (+ Diskette) DM 68,–
3627 GEM: Die VDI-Bibliothek (+ Diskette) DM 68,–
3673 ATARI ST Grafikbuch (+ Diskette) DM 68,–
SYBEX
– die guten Seiten Ihres Computers
Sybex Verlag GmbH
Vogelsanger Weg 111
4000 Düsseldorf 30
Telefon: 0211/618020
Mailbox: 0211/614731

neue Betriebssystem notwendig ist. EUMEL ist weitaus komplexer als MS-DOS (geschweige denn GEM-DESKTOP) und ähnelt in vielen Punkten UNIX. Wer wollte da an einem Tag durchstarten?

Unsere EUMEL-Station arbeitet jetzt so: Bei eingesteckter ROM-Cartridge können GEM und alle ST-Programme ganz normal aufgerufen werden, wir verwenden dazu die Partitions C und E der Festplatte. Partition D nimmt für EUMEL 10 MByte ein und ist unter GEM nicht ansprechbar. Der Start von EUMEL geschieht durch einen Doppelklick auf das ROM-Icon, dort erscheint „EUMEL.PRG", und mit einem weiteren Doppelklick fährt man EUMEL von der Festplatte hoch (Bild 3). Leider kann man von EUMEL aus nicht mehr zum Desktop zurück – dazu ist ein Reset erforderlich.

Doch nun zurück zu EUMEL. Das Betriebssystem besitzt gewissermaßen drei Benutzeroberflächen. Der **Supervisor** meldet sich als „Platzanweiser" zum Arbeitsbeginn. Hier kann man einen Überblick über alle laufenden Tasks gewinnen („task info" in Bild 4), eine neue Task einrichten („begin") oder sich an eine alte ankoppeln („continue"), wenn diese verlassen wurde oder einen Batch-Auftrag bearbeitete. Mit der Taste „SV" (auf dem Atari: F1) wird der Supervisor jederzeit aufgerufen, auch wenn gerade ein Programm läuft. Da EUMEL seinen Betriebszustand ständig auf Festplatte konserviert, ist es nicht notwendig, einmal insertierte Programme erneut zu laden. Vielmehr landet man nach dem Ankoppeln einer Task immer dort, wo man zum Schluß aufgehört hat.

In der Regel ist dies die **Monitorebene**, die den Benutzer mit „gib kommando" begrüßt. Diese Ebene ähnelt etwas der MS-DOS Oberfläche, ist aber viel raffinierter gebaut. Neben den Standardfunktionen – Taskinhalt zeigen („list"), Dateien verschicken („save") oder holen („fetch"), kopieren („copy") oder löschen („forget") – gibt es ein besonderes Bonbon: Als Kommandobefehle stellt die Monitorebene nahezu den gesamten Sprachumfang von ELAN zur Verfügung. Während unter MS-DOS jeder Anwender mindestens zwei Sprachen lernen muß (Programmiersprache und Monitorsprache), reicht es für alle EUMEL-Benutzer völlig aus, eine einzige Sprache zu kennen: ELAN.

```
..................... list .....................   Zeile 1
.................. printer.nec.p5 ...............   Zeile 117
      THEN is nlq := FALSE;
      ELSE is nlq := std quality = "nlq"
   FI;
   x steps := x step conversion ( x size );
   y steps := y step conversion ( y size );
   y steps := (y steps DIV 30) * 30;
   modification bits := 0;
   out (""24""27""64"");                       (* Reset des Druckers          *)
   out (""27"R"0"");                           (* Amerikanischer Zeichensatz *)
   out (""27"C" + code (y steps DIV 30)); (* Formularlaenge            *)
   out (""27"x"0"");                           (* Entwurfsqualität           *)


. x start : param1
. y start : param2
.
  open page :
    x start := 0;
    y start := 0;
    x rest  := 0;
    out (""13"");
```

Bild 2: Druckeranpassung einmal anders: Ein ELAN-Programm für den NEC

```
**** ATARI-ST SWord, Release 1.3 ****
@ Alle Rechte bei BICOS COMPUTER GmbH
Harddisk-Version
ROM-MODUL SN 29-01-87 : 1113-026-MeinST

Testversion - Redaktion ST Computer


Urlader wird vom Hintergrund geladen...

        E U M E L - Vortest
      VERSION :101709 201086

Terminals : 0, 1, 2, 15, 30, 31,
RAM-Groesse (gesamt): ...820 KB
Pufferbereich: ..........694 KB
Hintergrund-Speicher: ...10568 KB

Speichertest :

*************
****
```

Bild 3: Statusinformationen beim „Hochfahren"

```
SUPERVISOR
    SYSUR
        Kanal 2
        SCHEDULER
        DOS
        PRINTER
            -
        configurator
        OPERATOR
            shutup
        ARCHIVE
UR
    PUBLIC
        Werkzeuge II
        Hamster
        std
        Drucker
        texte
        zwei
        dasys
```

Bild 4: Ein Überblick aller Tasks. Beachten Sie bitte die Schachtelung, die Väter, Söhne und Enkel angibt.

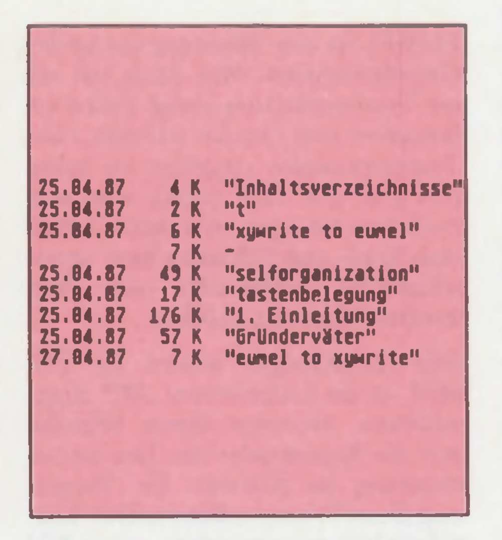
Mit diesem Prinzip ist die Erstellung von Batch-Aufträgen ein Kinderspiel. Ein Beispiel mag dies erläutern. In Task 1 befindet sich ein Text, für den der Zeilen- und Seitenumbruch vorzunehmen ist. Anschließend soll der fertige Text ausgedruckt werden. Aus der Supervisorebene loggen wir uns in Task 1 ein und geben folgendes Kommando: »autoform („dateiname");autopageform; print«. Damit wird der Text automatisch umgebrochen und gedruckt. Da unser Text sehr lang ist, verlassen wir die Task durch die Taste SV (Supervisor) und begeben uns in Task 2. Dort wird ein Programm mit dem Kommando »insert („programmname")« übersetzt. Auch diese Prozedur nimmt viel Zeit in Anspruch, deshalb verlassen wir auch hier die Task und arbeiten schließlich in Task 3, wo ein Text getippt werden soll. EUMEL erledigt nun drei Jobs gleichzeitig: Zum einen wird der Text von Task 1 bearbeitet und zur Spool-Task geschickt. Zum anderen ist Task 2 mit der Abarbeitung des Programms beschäftigt. Eventuelle Fehlermeldungen werden automatisch in ein „Notizbuch" geschrieben, das zu jeder Zeit einsehbar ist. Und schließlich tippen wir in Task 3 an einem Text, ohne von den beiden anderen Aufträgen gestört zu werden. Bequemer geht es wohl nicht.

Ein weiteres Feature der Monitorebene soll nicht unerwähnt bleiben: Der „Thesaurus". Auch dazu ein Beispiel. Unser Taskinhalt soll dargestellt werden; dazu benötigen wir den Befehl „list". In der Monitorebene erscheint

nun ein Fenster, das alle Dateinamen aufführt (Bild 5). Das wäre nichts besonderes, wenn dieses Fenster kein Editorfenster wäre. Wir können nämlich mit den Editorkommandos beliebig durch den Text scrollen, Teile markieren, in eine andere Datei kopieren und/oder ausdrucken lassen. Das Thesaurusprinzip wird auch bei größeren Operationen geschickt genutzt: Angenommen, ein Benutzer will eine Auswahl von Dateien aus der Vatertask holen. Er gibt dazu ein: »fetch (SOME public, public)« und es werden zunächst alle Dateien aufgeführt. Mit den Editorfunktionen Zeile oder Bereich löschen werden nicht benötigte Dateinamen gelöscht, und sobald das Fenster verlassen wird (ESC-q), werden genau diejenigen Dateien in die Task geholt, deren Name zum Schluß noch im Fenster stand.

```
testtext
testtext.p
schöntext
Inhaltsverzeichnisse
t
xywrite to eumel
eumel to xywrite
selforganization
tastenbelegung
1. Einleitung
Gründerväter
selforganization.p
zeichensätze
tastatur
f
1. Einleitung.p
```

Bild 5: Der Taskinhalt mit dem Kommando „list“



```
25.04.87    4 K  "Inhaltsverzeichnisse"
25.04.87    2 K  "t"
25.04.87    6 K  "xywrite to eumel"
            7 K  -
25.04.87   49 K  "selforganization"
25.04.87   17 K  "tastenbelegung"
25.04.87  176 K  "1. Einleitung"
25.04.87   57 K  "Gründerväter"
27.04.87    7 K  "eumel to xywrite"
```

Bild 6: Datenmüll auf der Diskette

Das Archiv

Zur dauerhaften Speicherung von Dateien oder Programmen wird auch beim Betriebssystem EUMEL auf Disketten zurückgegriffen. Es empfiehlt sich insbesondere bei fast belegter Festplatte, einige Dateien auszulagern, da ansonsten die Arbeitsgeschwindigkeit stark nachläßt. EUMEL unterstützt alle gängigen Floppyformate (8, 5,25 und 3,5 Zoll), aber das Aufzeichnungsformat ist weder Atari- noch MS-DOS-kompatibel. Das Archiv wird in der Regel aus der Monitorebene aufgerufen. Zunächst erfolgt die Reservierung aus einer Task mit »archive („archivname“)«. Sollte im Multiuser-Betrieb gerade ein anderer Benutzer auf die Diskettenstation zurückgreifen, wird eine Fehlermeldung ausgegeben. Dateien und Programme können einzeln »save („dateiname“,archive)« oder mit dem Thesaurus »save (SOME myself, archive)« gespeichert bzw. geholt (analog mit „fetch“) werden. Vor ungewolltem Überschreiben schützt eine Sicherheitsabfrage. Das Archiv wird schließlich für andere Benutzer mit »release (archive)« freigegeben.

Zwei Besonderheiten müssen erwähnt werden. Zum einen überschreibt EUMEL nicht immer die alten Dateien. Vielmehr werden diese als gelöscht gekennzeichnet (das sieht man auf Bild 6 an dem Bindestrich), was die negative Auswirkung hat, daß oftmals eine Diskette bereits voll zu sein scheint, obwohl die tatsächlich vorhandenen Dateien nur sehr wenig Platz in Anspruch nehmen. In diesem Fall müssen alle Dateien in die Task geholt, das Archiv gelöscht und die Dateien zurückgeschrieben werden. Zum anderen ist, wie gesagt, das EUMEL-Aufzeichnungsformat weder zu MS-DOS noch zum Atari ST kompatibel. Um dennoch alte Texte zu überspielen, existiert ein (mitgeliefertes) Transferpaket von Frank Klapper, mit dem unter MS-DOS formatierte Disketten gelesen und beschrieben werden können. Mit den neuen Befehlen war es auch möglich, ASCII-Files auf Atari ST-Disketten zu lesen bzw. zu schreiben – kein Wunder, da das Aufzeichnungsformat von MS-DOS und TOS fast identisch ist.

Der Editor

Der EUMEL-Editor ist zentral für das gesamte Betriebssystem. Mit ihm werden nicht nur Texte bearbeitet, sondern auch alle ELAN-Programme und Druckvorlagen für die Dateiverwaltung. Die am häufigsten verwendeten Tasten bei der Editierung sind zunächst die Cursortasten, die ESC-Taste, die „Verstärkungstaste“ HOP und die beiden Tasten RUBOUT und RUBIN. Die letzten drei Tastaturbezeichnungen kennt der Atari ST nicht. In unserer Konfiguration wurde folgende Umbelegung vorgenommen:

- "Insert" –> "Mark"
- "Clr/Home" –> "HOP"
- "Help" und "Delete" –> "RUBOUT"
- "Undo" –> "RUBIN"
- "F1" –> "Supervisor"
- "F2" –> "Weiter"
- "F3" –> "Stop"

Nach dem Aufruf des Editors aus der Kommandoebene mit "edit („Dateiname“) erscheint das in Bild 7 wiedergegebene Fenster. Die erste Zeile informiert über die gesetzten Tabulatoren, den Dateinamen, den Einfüge- oder Überschreibmodus und schließlich die aktuell bearbeitete Zeile.

Mit den beiden Tastaturkombinationen ESC-1 und ESC-9 läßt sich an den Anfang oder das Ende des Textes springen. Die HOP-Taste wirkt für die Cursorbewegungen als „Verstärkung“: Mit HOP-Cursorrichtung springt man an den Anfang oder das Ende der Zeile bzw. blättert im Text auf- oder abwärts. Einfaches Löschen von Buchstaben erfolgt mit RUBOUT; mit RUBIN kann der Einfügemodus ein- oder ausgeschaltet werden, so wie bei 1st Word Plus. Auch diese beiden Tasten besitzen in Kombination mit HOP eine Verstärkungsfunktion: HOP-RUBOUT löschte eine Zeile rechts vom Cursor. Mit HOP-RUBIN kann äußerst einfach zusätzlicher Text eingefügt werden: Ab der aktuellen Cursorposition verschwindet der restliche Text und ausreichend Platz für die neue Texteingabe wird freigegeben. Die Anzeige REST in der Titelzeile erinnert daran, daß noch ein Resttext existiert. Dieser erscheint nach einem neuerlichen Betätigen der beiden Tasten HOP und RUBIN. Wer bereits mit VIZAWRITE gearbeitet hat, wird dieses Prinzip kennen. Das wortweise Springen im Text ist beim EUMEL-

Editor mit den Tasten ESC-Cursor links/rechts implementiert.

Der Tabulator ist gleichermaßen Zahlen- und Texttabulator. Er wird mit der TAB-Taste angesprochen und richtet Ziffern automatisch rechtsbündig aus. Besondere Markierungen im Text werden übrigens nicht hinterlassen, vielmehr gilt bei allen Einrückungen und Tabellen das Prinzip „What you see is what you get" – obwohl die Textverarbeitung sonst mit Klartextbefehlen arbeitet.

Daß Blöcke markiert, verschoben, dupliziert und unter anderem Namen in den Hintergrund geschrieben werden können, ist bei EUMEL selbstverständlich. Alle Blockoperationen laufen mit der Taste MARK, die zeichengenaues Markieren von Textausschnitten erlaubt. Nicht so selbstverständlich sind jedoch die Lernfähigkeiten des EUMEL-Editors. So können nicht nur einzelne Tasten mit Floskeln vorbelegt werden, vielmehr existiert auch die Möglichkeit, ganze Kommandos mit einem Tastendruck zu programmieren.

Das ist auch unbedingt notwendig, denn das Portabilitätsprinzip (jedes Terminal ist anschließbar) verlangt, daß in einem Text keine Steuerzeichen vorkommen. Somit muß für alle Textattribute (fett, kursiv, unterstrichen, indiziert) eine Klartextanweisung eingefügt werden. Diese Anweisungen sind auch für die Textkosmetik unerläßlich.

Textkosmetik und viele Anweisungen

Zur Textgestaltung existieren auf gängigen Computersystemem zwei unterschiedliche Prinzipien. Das eine ist derzeit in aller Munde und heißt „What you see is what you get" („WYSIWYG"). Damit ist gemeint, daß unterschiedliche Textattribute, Schriftgrößen und -arten genau so auf dem Bildschirm erschienen, wie der Drucker sie ausgibt. Für den Atari ST ist SIGNUM ein Textprogramm, das ohne Ausnahmen in diesem Modus arbeitet. Die andere Möglichkeit sind Klartexteditoren, bei denen mit einer eigenen Programmiersprache die Textformatierung erfolgt. Ein Musterbeispiel stellt TEX dar, das jetzt auch für den Atari ST lieferbar ist.

```
..^.....RUBIN.........^... schöntext .....^..................... Zeile 33
believed till some time ago that it would be possible to introduce the theore
founded concept of scientific rationality; this was a construction which has
given up now.#type ("prop")##linefeed (2.0)##end#

In our opinion the theory of selforganizing systems opens up new
constructive possibilities at three decisive points, which simultaneo-
usly relieve the observer from making problematical predecisions.

a) This theory adopts factually and methodologically a new appro-
   ach to the problem of the relationship between science and its
   environment. The point of departure for the approaches til now
   has been the concept of factual alternatives between scientific
   autonomy and heteronomy, between internal and external deter-
   minants, etc.. Systems theory does not see any alternatives in
   these distinctions. It is not the case that scieoce is either
   autonomous (selfregulated) or heteronomous (externally control-
   led), it can increase its autonomy through its heteronomy or
   weaken it. These distinctions are features which structure the
   system-environment relationship maintained by the system. The
   system itself demarcates itself from its environment. It regulates
   this difference along the margin or boundary thus created (#count#).#foot#
(#value#)#type ("prop15")##linefeed (1.0)# Boundary conditions in the classic
systems' variables. In the case of mathematical description they do not resul
```

Bild 7: Ein typisches Editorfenster mit Tabulatoren, Statusinformationen und Einrückung.

EUMEL ist eine Mischung aus beiden Grundprinzipien. Man kann mit seiner Textverarbeitung einige Zeilen unformatiert zum Drucker schicken, ohne Textkommandos eingefügt zu haben (das geht mit TEX nicht), aber man muß Anweisungen verwenden, sobald man Kopf- und Fußzeilen bzw. unterschiedliche Zeichensätze oder Textattribute einsetzen möchte.

Alle Anweisungen werden bei EUMEL in die Lattenkreuze „#" eingeschlossen, zwischen diesen befinden sich die Kommandos zur Druckeransteuerung im Klartext. Ein Beispiel mag dieses Prinzip erläutern. Der Textanfang könnte so aussehen, wie in Bild 8 wiedergegeben.

Die erste Zeile weist den Drucker an, in der Schriftart „prop" auf einer Zeilenbreite von 14 cm zu drucken. Die zweite Zeile gibt einen doppelten Zeilenabstand vor. Die dritte Zeile legt das Prozentzeichen als Marker für die Seitennummer fest und setzt den Zähler auf Eins. Die vierte bis siebte Zeile definieren eine Kopfzeile, deren erste Zeile die zentrierte Seitennumer enthält und der dann eine Leerzeile bis zum Textanfang folgt. Die achte und neunte Zeile geben an, daß der Text horizontal und vertikal im Blocksatz ausgedruckt werden soll.

Das allgemeine Prinzip der Klartextkommandos gilt auch für die verschiedenen Textattribute. Soll ein Wort

```
.......................... selforganization_........................ Zeile 11
#type ("prop")##limit (14.0)#
#linefeed (2.0)#
#pagenr("%",1)#
#head#
#center#- % -

#end#
#block#
#pageblock#
Science as selforganizing system

Draft discussion paper                    Bielefeld 1987



The approach of the theory of selforganizing systems

In general this theory functions with less preliminary decisions than
the theories we have mentioned up to now because its ability to
structure is greater:

It is less descriptive because it is more constructive. n the case of
```

Bild 8: Der Textanfang mit Druckeranweisungen.

unterstrichen werden, so ist einzugeben: #on(„u")#unterstrichen#off(„u")#. Da es sich bei allen Steuerkommandos ausschließlich um Textstrings handelt, ist die Portabilität von einem Terminal (oder Computer) zum anderen gewährleistet. Ein weiterer Vorteil liegt auf der Hand: Alle Anweisungen lassen sich mit der Suchoption finden und ersetzen oder können schnell neuen Erfordernissen angepaßt werden. Ein Beispiel: Wer auf einem „WYSIWYG"-System den äquidistanten Zeichensatz in einen proportionalen ändern möchte, muß in der Regel den gesamten Text neu formatieren. Bei EUMEL reicht es völlig aus, die „type"-Anweisung am Anfang des Textes zu modifizieren – eine Arbeit von Sekunden. Allerdings erkennt man jetzt auch, warum der EUMEL-Editor unbedingt lernfähig sein muß: Ohne Tastenvorbelegung müßte für jede Kosmektikanweisung ein Wort eingetippt werden.

Mit den Anweisungen lassen sich alle Formatierunsmöglichkeiten ausschöpfen, die auch Buchdrucker auf ihrer Satzmaschine vorfinden. Tabelle 1 liefert einen kleinen Ausschnitt aller Funktionen:

Tabelle 1: Textformatierungen mit EUMEL

- Alle vorhandenen Schrifttypen und Textattribute dürfen in einer Zeile beliebig oft gewechselt werden.
- Horizontaler und vertikaler Blocksatz mit allen Schriftbreiten und -höhen, Stauchung oder Streckung einer Seite.
- Präzise Justierungsmöglichkeiten für unterschiedliche Textbreiten auf einer Seite, da alle Angaben in Millimetern eingegeben werden.
- Zeilenabstände können an jeder Stelle im Text in Millimeterschritten variiert werden. Die Ausdrucklänge hängt nur von der vorgegebenen Seitenlänge in ab.
- Echte Spaltenverarbeitung- und formatierung (Zeitungsdruck); Blocksatz auch für zentrierte Spalten.
- Frei gestaltbare Kopf- und Fußzeilen über mehrere Zeilen, für gerade und ungerade Seiten alternierend; bei Bedarf abschaltbar.
- Fußnotenverwaltung mit automatischem oder manuellem Umbruch bei sehr langen Fußnotentexten.
- Formelsprache für Tabellen und Spalten, Zwischenräume bei Tabellen können automatisch ausgefüllt werden, Tabellendruck auch in Proportionalschrift.
- Formelsprache für naturwissenschaftliche Anwendungen.
- Querverweise auf Textteile oder Fußnoten mit Platzhaltern.

Die Formatierungsanweisungen erscheinen, wie gesagt, ausgeschrieben auf dem Terminal. Dies zeigt Bild 9 als Hardcopy. Man sieht, daß durch die vielen Anweisungen der Überblick auf dem Monitor zwangsläufig zurücktritt. Dazu kommt als weiteres Manko, daß alle Anweisungen den Platz eines Wortes einnehmen. Hat man beispielsweise zweimal innerhalb einer Zeile unterstrichen, so rollt der Text über den rechten Bildschirmrand hinweg und ist – bei einer Zeilenbreite von 65 Zeichen auf dem 80 Zeichen-Monitor – nur durch seitliches Blättern ganz zu überschauen.

```
.....RUBIN................ schontext .............................. Zeile 18
oretical definitions which phenomena form a part of the scientific
subject matter. An example explains this point: a descriptive theory
of life will allow animals and plants as subject matters with such
definitions. As far as the constructive theory of life (as for instan-
ce in the case of #ib (1)#Eigens#ie (1)# theory of selforganization matter)
is concerned, animals and plants as well as the entire range of life on
earth known to us are an (accidental) solution of the theory and
other theories are permissible. In #ib (1)#scientific research#ie (1)# it has
to now been largely the task of the observer to make definitions regarding
the subject matter and to #u#differentiate#e# between "internal" (system)
and "external"(environment). The #on("u")#theory of
selforganization#off("u")#
permits
us to look upon these #on("b")#definitions#off("b")# as social processes
which are
an achievement of the subject area itself. The fact that this shift does
not result in an arbitrary relativism or descriptive chaos ("in that
case everthing could be science or life") is a result of the limita-
tions posed by selforganization itself (#count#).#foot#
(#value#)#type ("prop15")##linefeed(1.0)# The theoretical progress of the tra
theories is always only relative and gradual; e.g. in our model we have to us
language which regulates the use of words like #ib (1)#science#ie(1)#, resear
```

Bild 9: Steuerzeichengewirr auf dem Bildschirm – ein Minuspunkt der Textverarbeitung

Im Gegensatz zu TEX verfügt EUMEL nicht über einen Modus, bei dem man sich den endgültigen Ausdruck auf dem Schirm anschauen kann: Dazu benötigt man immer den Graphikmodus eines Terminals und dies widerspräche dem Portabilitätsprinzip. Somit ist der Anwender auf zahlreiche Ausdrucke angewiesen, bis ein Text endgültig „steht". Nach meinen Erfahrungen steigt die Zahl der Vorabdrucke mit der Zahl der Anweisungen an – auch ein EUMEL-Experte kann nie sicher sein, daß der Text so gedruckt wird, wie er es wünscht.

Zeilen- und Seitenumbruch

Wie der Zeilen- und Seitenumbruch vorgenommen wird, zeigt Bild 10. Mit „lineform" oder „autoform" sorgt EUMEL dafür, daß die Zeilen den verwendeten Schrifttypen entsprechend umgebrochen werden. Während „lineform" dem Benutzer Trennvorgaben anbietet (mit dem besten Algorithmus, den ich je gesehen habe), verwendet „autoform" die dem Trennalgorithmus entsprechenden Vorgaben ohne Benutzereingriffe und schreibt (zum Nachschauen) alle Trennungen in das „Notizbuch". Ein Ausnahmelexikon für falsche Trennvorgaben kann eingeladen und die Trennbreite kann zwischen 4 und 20 Prozent der Zeilenbreite festgelegt werden. Für den Seitenumbruch gilt das gleiche Prinzip: „pageform" liefert dem Benutzer Vorgaben zum Seitenumbruch und „autopageform" nimmt diesen automatisch vor. Erst mit dem Seitenumbruch liegt der Text in seiner endgültigen Fassung vor: Die Fußnoten befinden sich dann unten auf der Seite, und sofern man

Macros (z. B. für einen Briefkopf) benutzt hat, sind diese durch den Klartext ersetzt worden.

Mit dem Programm „index“ können Stichwort- und Inhaltsverzeichnisse erstellt und sortiert werden. Dazu müssen natürlich die entsprechenden Überschriften oder Worte im Text selbst markiert sein. Auch dies geschieht mit Klartextanweisungen, in Bild 9 sind dies die Anweisungen #ib(1)# und #ie(1)#. Eine Zusammenfassung von Kapitelüberschriften und Stichworten liefert die Routine „outline“, mit der eine Datei erzeugt wird, die die Kapitel und zugehörige Stichworte numeriert und einrückt. Damit kann ein Text auf logische Konsistenz überprüft werden.

Soweit die wesentlichen Möglichkeiten der Textgestaltung. Nicht alle Fähigkeiten können hier aufgeführt werden, denn EUMEL ist ein Textsystem, bei dem die Regel gilt: Es gibt nichts, was es nicht gibt. Und wer tatsächlich eine Kleinigkeit vermißt, kann mit ELAN auch exotische Sonderwünsche ohne großen Programmieraufwand realisieren. Doch dazu mehr in der nächsten Folge.

(Michael Spehr)

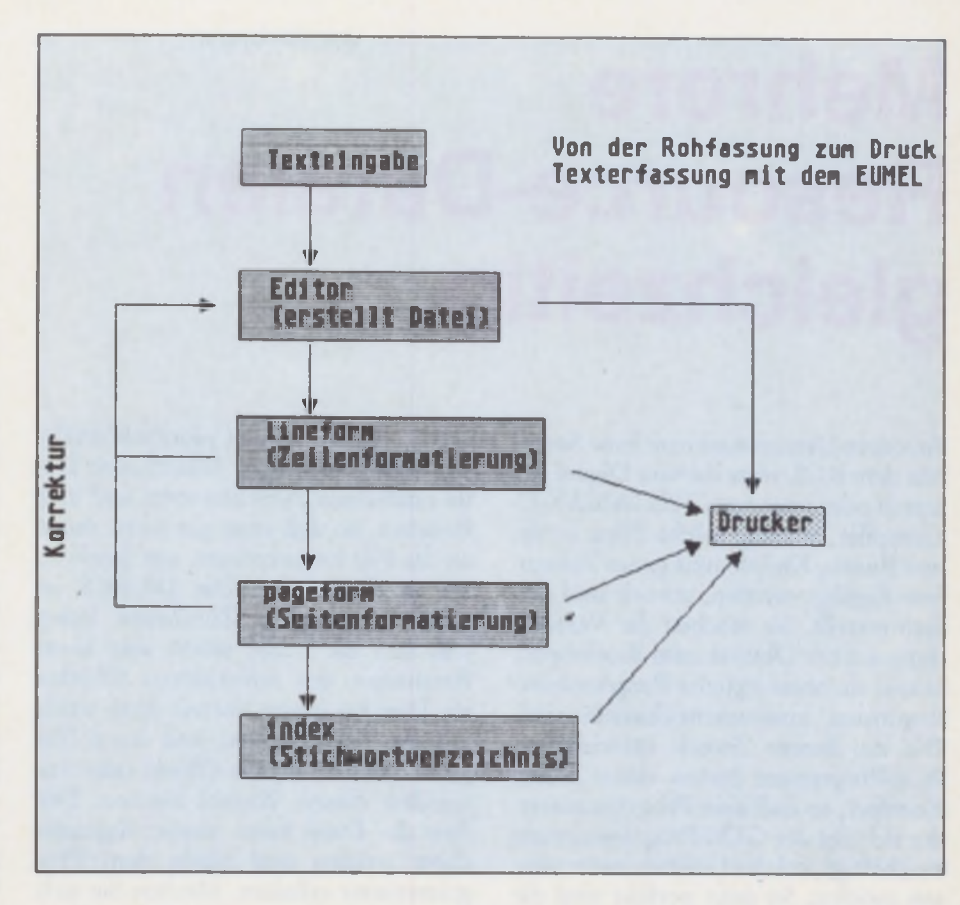


Bild 10: So läuft der Zeilen- und Seitenumbruch ab

Mehrere Resource-Dateien gleichzeitig

Resource-Dateien sind eine feine Sache. Mit dem RCS, sei es das von Digital Research oder jenes vom MEGAMAX-C-Compiler, werden solche Files, in denen Boxen, Knöpfe und ganze Formulare abgelegt werden, schnell und einfach erstellt. Sie machen die Verwendung solcher Objekte zum Kinderspiel, zumal sie ohne jegliche Programmierkenntnisse zusammenzubasteln sind. Die zu diesem Zweck entwickelten RCS-Programme bieten einen hohen Komfort, so daß kein Programmierer, der sich mit der GEM-Programmierung beschäftigt, solche Utilities mehr missen möchte. So ganz perfekt sind die Programme allerdings nicht, was man schon bei den ersten Versionen des DR-RCS sehen konnte. Beim Einladen eines Resource-Files, in dem man ein Icon konstruiert hatte, stürzte das Programm sang- und klanglos ab. Durch ein paar kleine Überarbeitungen hat man diese Probleme in den Griff bekommen. Trotzdem kann man mit diesen Programmen nicht uneingeschränkt arbeiten. Ein wichtiger Hinderungsgrund ist die maximale Größe einer RSC-Datei. Die Dateigröße darf den Wert von etwa 30 kB (Augenmaß) nicht überschreiten. Offenbar wurde beim Programmieren auf die besondere Speicherausstattung eines Atari ST keine Rücksicht genommen.

Wer hier und da mal ein kleines GEM-Programm schreibt, der wird nie in Schwierigkeiten kommen, daß plötzlich die RSC-Datei nicht mehr ausreicht. Aber auch da gibt es Tücken. Bei meiner Arbeit verwende ich grundsätzlich zwei RCS-Systeme, das von MEGAMAX und das von DR. Das MEGAMAX-Programm zeichnet sich durch einfache Bedienung und – das schätze ich besonders – durch den sehr schönen eingebauten Iconeditor aus. Durch häufige Änderungen an einer Datei kann es aber vorkommen, daß – aus welchen Gründen auch immer – die RSC-Datei plötzlich defekt ist. Beim Laden einer fehlerhaften Datei erscheinen zwischen zwei und vier Bomben, so daß man gar nicht mehr an das File herankommt, um den Fehler zu eliminieren. Das DR-RCS ist nicht so pingelig. Zumindest laden läßt sich die Datei, stürzt aber beim Bearbeiten des fehlerhaften Objekts ab. Dies hat einen Vorteil: Man weiß, wo der Fehler steckt und kann (im günstigsten Falle) das Objekt oder zumindest dessen Wurzel löschen. Der Rest der Datei kann wieder abgespeichert werden und bleibt dem Programmierer erhalten. Machen Sie sich bei häufigen Änderungen immer eine, oder besser, mehrere Sicherheitskopien. Ob eine Datei auch wirklich richtig abgespeichert wurde, können Sie meistens nicht am Programm erkennen, das die Source verwendet, sondern erst beim erneuten Bearbeiten mit dem RCS. Und wenn dann der Fehler entdeckt wird, ist es zu spät.

Kommen wir zum Hauptproblem bei umfangreicheren Programmen. Sie brauchen lediglich in Ihren Programmen mit einigen Icons zu arbeiten, schon steigt der Platzbedarf einer solchen Datei steil an. Noch gravierender ist es, wenn man sich eine kleine Tabelle aus Objekten (Rähmchen, Boxen, Überschriften usw.) konstruieren will, um dort Eingaben zuzulassen. Das kostet, über den Daumen gerechnet, in ansprechender Form gut 5 kB Speicher. Für den Atari ST ist das keine Frage, wohl aber für das RCS.

Ganz gemein wird es, wenn Sie in Bereichen zwischen 20 und 30 kB pro RSC-Datei arbeiten. Dann kann es vorkommen, daß die Datei einwandfrei abgespeichert wird, aber wegen Speicherplatzmangels nicht mehr wieder eingeladen und bearbeitet werden kann. Bevor es soweit kommt, sollte man seine Dateien in zwei etwa 10 bis 20 kB große Files aufsplitten.

Wie man die Dateien in den Speicher bekommt, ist bereits durch zahlreiche Artikel bekannt. Zum einen kann man eine Datei mit der Endung „.C" vom DR-RCS ausgeben lassen, um sie anschließend durch „#include" vom Compiler mit einbinden zu lassen. Beim Programmstart braucht dann lediglich eine kleine Routine diese Daten an die aktuelle Bildschirmauflösung anzupassen. Die zweite Möglichkeit ist wesentlich bequemer. Durch Aufruf der GEM-Funktion rsrc_load() wird die „.RSC"-Datei in den Speicher gelesen und automatisch angepaßt. Dies wird wohl der Normalfall sein. Hat man nun zwei Dateien, so benötigt man auch zwei rsrc_load()-Aufrufe. So einfach, wie dies zunächst aussieht, ist es aber nicht. Denn der zweite Aufruf überschreibt zwar nicht die erste Datei, aber man hat nur Zugriff auf das zuletzt geladene Resource. Es wäre natürlich möglich – je nachdem, welche Objekte man aus den Dateien benötigt – die eine oder andere RSC-Datei zu laden und anschließend, wenn diese Informationen nicht mehr gebraucht werden, den belegten Speicherplatz mit rsrc_free() wieder freizugeben.

Das Programm wäre natürlich wegen der dauernden Diskettenzugriffe viel zu langsam. Es gibt aber eine wesentlich bessere Lösung. Dazu ist etwas Wissen über die Interna von GEM nötig. Man muß, wie jeder GEM-Programmierer weiß, unter anderem ein Array namens „global" für das Betriebssystem bereitstellen. In diesem Speicher legt GEM nach dem rsrc_load()-Befehl verschiedene Daten ab. Uns interessiert der Eintrag in global[5] und global[6]. Hier findet man nach dem erfolgreichen Aufruf einen Zeiger auf die Baumstruktur. Da dieser Zeiger 4 Byte beansprucht und das Array Integerformat (2 Byte) besitzt, müssen zu diesem Zweck zwei Einträge herhalten. Das Low- und High-

Word sitzen an der richtigen Stelle, so daß der geübte C-Programmierer nur noch eine Typumwandlung von Integer in Objektzeiger durchführen muß, um ihn zu verarbeiten. Diesen Zeiger müssen wir zwischenspeichern, da beim nächsten Funktionsaufruf von rsrc__load() dieser Wert überschrieben wird. Der belegte Speicherplatz jedoch und alle dazugehörigen Informationen bleiben unangetastet. Das zweite RSC-File wird irgendwo anders plaziert und der Zeiger darauf ist wieder im global-Array abgelegt. Für jede RSC-Datei brauchen wir also eine Variable für einen Objektzeiger.

Nach dem letzten Laden sind nur die aus dieser Datei stammenden Informationen erreichbar. Das liegt an dem „getarnten" Zeiger in global. Jetzt kommt der Clou an der Sache! Wird dieser Zeiger jetzt auf den ersten Wert umgeändert, so ist wieder Datei Nummer eins ansprechbar, beziehungsweise deren Bäume und Objekte. Wie bei „älteren" Rechnern, deren Prozessor mehr Speicherplatz zur Verfügung hat, als er eigentlich verarbeiten kann, wird durch Hin- und Herschalten zwischen verschiedenen Speicherbereichen ausgewählt. So dürfte das Problem übrigens auch bei 1st__Wordplus gelöst sein, da dort ebenfalls mit mehreren (5) RSC-Dateien gearbeitet wird, die getrennt auf der Diskette vorliegen.

Die Routine, die das Umschalten übernimmt, kann man Listing 1 entnehmen. Vorausgesetzt wurde hier, daß in rsc__ptr[] bereits die Pointer nach dem Laden abgespeichert wurden. In diesem Beispiel gehe ich von 5 Dateien aus, die von 0 bis 4 durchnumeriert sind. Man kann dann durch den Aufruf von

rsc__file(2);

auf Datei 2 umschalten. Wer sich schon etwas näher mit C und den untrennbar damit verbundenen Pointern befaßt hat, der wird auch die Zuweisung

ptr = (OBJECT ★★) &global[5];

verstehen. „&global[5]" liefert die Adresse dieses Elements (des gesuchten Pointers), das nun jedoch in einen Speicher eines Zeigers umgewandelt werden muß. Deshalb die zwei Sternchen, denn ein Stern würde lediglich aussagen, daß diese Adresse der Zeiger selbst auf eine Objekt-Struktur wäre. Die Zuweisung danach ist dann auch zu verstehen, wenngleich auch sie zusammen mit der obigen Zeile zu einer einzigen verschmolzen werden kann. Aber ich finde, es ist auch so schon kompliziert genug.

Das Auslesen beim Laden der RSC-Dateien entnehmen Sie bitte Listing 2.

Dazu bleibt nur zu sagen, daß in der Variablen „file__pfad" der aktuelle Pfad zu den Resource-Dateien angegeben werden kann (z. B. mit der Gemdos-Routine Dgetpath oder im Source bestimmt). Die Dateinamen sind im rsc__dat[]-Array eingetragen. Nun viel Spaß beim Programmieren!

(Dirk Schaun)



```
Listing 1:

OBJECT *rsc_ptr[5];/* Ptr auf Beginn einer RSC-Datei */

rsc_file(n)
register int n;
{
    register OBJECT **ptr;

    ptr = (OBJECT **) &global[5];
    *ptr = rsc_ptr[n];
}
```

```
Listing 2:

#include "datei0.h"
#include "datei1.h"
#include "datei2.h"
#include "datei3.h"
#include "datei4.h"


char rsc_dat[5][14] =
{
"datei0.rsc"
"datei1.rsc"
"datei2.rsc"
"datei3.rsc"
"datei4.rsc"
};

...
int abbruch, i;

for(abbruch = i = 0; i < 5; i++)
   {
       sprintf(filename, "%s%s", file_pfad, rsc_dat[i]);
/* Immer aktuellen Patch benutzen */
       if(!rsrc_load(filename))
           {
               abbruch = 1;
               break;
           }
       ptr = (OBJECT **) &global[5];    /* Ptr initialisieren */
       rsc_ptr[i] = *ptr;
   }
if(!abbruch)
   hier geht's weiter...
else
   Fehlerbehandlung...
```

Ausgewählte Kapitel der BASIC-Programmierung

BASIC erweist sich immer mehr als beliebteste Sprache in verschiedenen Bereichen der Computertechnik. Nicht nur der Heim-Programmierer benutzt sie am häufigsten, sondern auch in der Industrie und an Universitäten gewinnt BASIC an Ansehen. Die neuen Versionen dieser Interpretersprache sind so erweitert und leistungsfähig, daß es kaum Argumente für die Ablehnung gibt. Ein neues Indiz für die Bedeutung dieser Programmiersprache ist die neueste Version eines BASIC-Interpreters von dem renommierten Softwarehauses BORLAND, der mit seinem TURBO-BASIC für IBM-Rechner und kompatible seine erfolgreiche Serie weiterführt. Für die Atari-ST-Serie gibt es schon seit langer Zeit eine Reihe von BASIC-Versionen, die wegen ihrer Geschwindigkeit und ihrer Leistung mit anderen höheren Sprachen problemlos konkurrieren können. Wir haben unsere Serie dieser wichtigen Sprache gewidmet und versuchen, in diesem Raum für jeden ein wenig zu bringen. Wir wollen also Einsteiger und Aufsteiger gleichermaßen ansprechen und hoffen, daß es gelingt.

Im ersten Teil dieser Serie beschäftigten wir uns mit der Ausgabe auf den Bildschirm und betrachteten eine Reihe von Befehlen, die die Ausgabe auf der Mattscheibe steuern. Niemand wird bezweifeln, daß dies eine wichtige Rolle bei der Programmierung spielt.

Eine andere wichtige Funktion, die uns immer verfolgen wird, ist die Eingabe. Dieser zweite Teil der Serie wird der Eingabe über die Tastatur gewidmet.

Eine Maus reicht nie aus

Neben Maus und GEM-Oberfläche gibt es beim Atari ST auch andere Eingabemöglichkeiten. Eine davon (und sicher eine der wichtigsten) ist die Tastatur. Sie wird nicht nur in Programmen, die unter TOS arbeiten, sondern auch bei Anwendungen, die unter GEM laufen, benutzt.

In BASIC gibt es verschiedene Funktionen, die die Eingabe über die Tastatur erlauben. Einige Befehle sind sozusagen standardisiert, sie gehören mehr oder weniger zu jeder BASIC-Version; andere sind Sonderfunktionen einer bestimmten Interpreter-Version. Da wir in dieser Serie die am meisten verbreiteten BASIC-Interpreter benutzen, nämlich die von GFA und OMIKRON, werden wir auch die Sonderfälle jeder Version besprechen.

Aus PUT wird INPUT

Um irgendetwas von der Tastatur zu lesen (zum Beispiel wenn der User die Taste 'A' drückt), muß der Atari ST jede Menge leisten. Ein in der Tastatur untergebrachter Prozessor (Coprozessor) verwaltet jede einzelne Taste und teilt dies dem Hauptprozessor jedes Mal mit, wenn sie gedrückt wurde. Der gibt dann dem Betriebssystem die Meldung weiter. Vom Betriebssystem her kann diese Information weiterverarbeitet werden; es sorgt dafür, daß sie gut bewahrt wird. Gleichzeitig steht sie zu irgendwelchen anderen Zwecken zur Verfügung. So kann man zum Beispiel die gewohnte Information auf dem Bildschirm darstellen, auf einem Drucker ausgeben oder als Parameter für eine andere Funktion benutzen, oder...

Wenn man sich gut im Betriebssystem des ST auskennt, kann man dieses kleine Beispiel (Taste A drücken und auf dem Bildschirm darstellen) ohne weiteres in Maschinensprache ausführen. Es soll Leute geben, die das mit Genuß tun. Für alle anderen, die sich noch nicht so vertraulich auf dieser Ebene bewegen, bietet ein BASIC-Interpreter sehr bequeme Befehle, die das Gleiche erreichen, ohne dabei Alpträume zu bekommen.

Der am häufigsten angewendete Befehl, um eines oder mehrere Zeichen über die Tastatur einzugeben, ist der Befehl INPUT. Seine Syntax ist bei GFA und OMIKRON gleich. Immer, wenn der Interpreter auf den Befehl INPUT trifft, wird der Ablauf des Programms gestoppt und auf eine (oder mehrere) Eingaben gewartet. Der Befehl INPUT ermöglicht die Eingabe von Zahlen oder Strings. Werden mehrere Eingaben hintereinander eingegeben, so müssen sie durch Kommata voneinander getrennt werden. Die Eingabe muß durch Drücken der Return-Taste abgeschlossen werden. Erfolgt eine erlaubte Eingabe, so wird das Programm fortgesetzt. Nun die genaue Syntax von INPUT:

INPUT A
Ermöglicht die Eingabe einer Zahl

INPUT A,B,C
Ermöglicht die Eingabe von drei Zahlen

INPUT A$
Ermöglicht die Eingabe eines Strings

INPUT A$,B$,C$,
Ermöglicht die Eingabe von drei Strings
Man kann die Eingabe von Zahlen und Strings, falls es notwendig ist, bunt vermischen:

INPUT A,B$,C,D,E$
Die erste, dritte und vierte Eingabe ist numerisch, die zweite und fünfte vom Typ String.

Wie man sehen kann, ist der Befehl sehr praktisch, um einem Programm verschiedene Parameter mitzuteilen. Es ist auch möglich, eine Eingabe mit einer Meldung zu versehen:

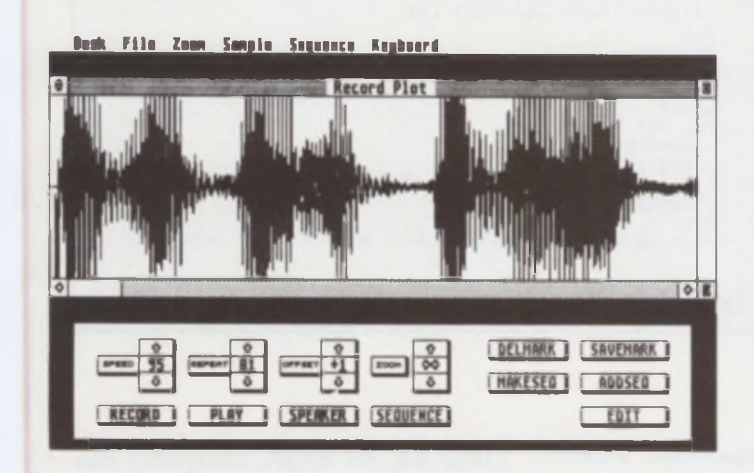
INPUT "Bitte geben sie jetzt Ihren Namen ein" ; Na$
Schließt man die Meldung mit einem Komma statt mit einem Semikolon ab, erscheint das ansonsten auftretende Fragezeichen nicht.

OMIKRON hat alle INPUT-Befehle um eine interessante Variante erweitert. Man kann den Cursor so steuern,

Der Unterschied ist deutlich...

G DATA

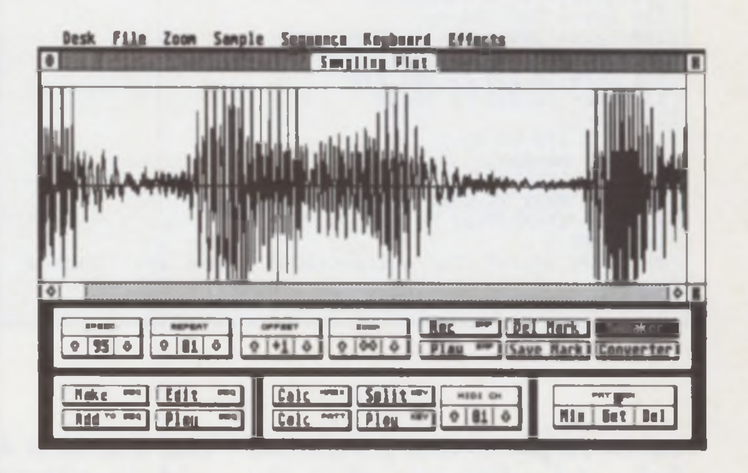
AS SOUND SAMPLER



- Einfaches Digitalisieren jedweder Töne (Musik, Sprache, etc.) in sehr guter Qualität
- Zahlreiche Editierfunktionen für digitalisierte Sounds mit dem integrierten Soundeditor
- In fertige Programme leicht einzubinden (GFA BASIC s. Demoprogramm, C, Assembler, usw.)
- Maximale Wandlungsfrequenz: 55 KHz, stufenlos regelbar
- Samplingzeit: ca. 1 Minute bei sehr guter Qualität
- Im Lieferumfang der Software: Schaltpläne, Platinenlayout, Bauanleitung und C Sourcecodes
- Bei Selbstbau der Hardware Bauteilekosten ca. DM 70,—

Software: **DM 79,—**
GFA BASIC Demo **DM 20,—**

AS SOUND SAMPLER II



- Alle Funktionen des AS SOUND SAMPLER.

Zusätzlich:

- Erweiterter Sound Editor
- Schnellerer Bildaufbau
- Komfortablere Benutzeroberfläche
- Digitale Verstärkung
- Automatisches Echo
- Mischfunktion
- Erweitertes Diskhandling
- Durch Softwaremodulation 40 % bessere Qualität
- GFA BASIC Demo Einbindung integriert
- PLAY und RECORD abbrechbar

Software: **DM 149,—**

Update für AS SOUND SAMPLER Besitzer gegen Einsendung der Originaldiskette und DM 70,— bar/Scheck

---

Fertig aufgebaute Hardware 8 Bit AD/DA Wandler anschlußfertig mit Netzteil
Für AS SOUND SAMPLER und AS SOUND SAMPLER II **DM 128,—**

Spezialhardware mit zusätzlichem Filter, Dynamikerweiterung, durchgezogenem Druckerport, Kontrolldioden, Druckerport gepuffert, regelbares Eingangsfilter
Für AS SOUND SAMPLER und AS SOUND SAMPLER II **DM 249,—**

...für alle ATARI ST

G DATA



Siemensstr. 16 · 4630 Bochum 1
TELEFONISCHE BESTELLUNGEN: 02325/60897

Schweiz: Senn Computer AG
Langstr. 21 · CH-8021 Zürich



```
100 ' Beispiel INPUT in OMIKRON
110 INPUT "Bitte geben Sie die Menge der Werte an ";Mg
120 DIM We(Mg)
130 FOR I=0 TO Mg-1
140      PRINT "Eingabe ",I+1;
150      INPUT We(I)
160      Summe=Summe+We(I)
170 NEXT I
180 MiweS= STR$(Summe/Mg)
190 Abst= INT(100/Mg)
200 Teil= INT((400-(Abst*Mg))/Mg)
210 CLS
220 DRAW 10,230 TO 400,230
230 DRAW 10,230 TO 10,20
240 X1=10
250 FOR I=0 TO Mg-1
260     Ms=Ms+1
270     FILL STYLE =2,Ms
280     PBOX X1,230 TO X1+Abst,230-We(I)
290     X1=X1+Teil
300 NEXT I
310 TEXT 10,290,"Das aritmetische Mittel ist :"+Miwe$
```

Listing 1

```
Input "Bitte geben Sie die Menge der Werte an ";Mg
Dim We(Mg)
For I=0 To Mg-1
  Print "Eingabe ",I+1;
  Input We(I)
  Summe=Summe+We(I)
Next I
Miwe$=Str$(Summe/Mg)
Abst=Int(100/Mg)
Teil=Int((400-(Abst*Mg))/Mg)
Cls
Defline 1,2,2,1
Line 10,230,400,230
Line 10,230,10,20
X1=10
For I=0 To Mg-1
  Ms=Ms+1
  Deffill 1,2,Ms
  Pbox X1,230,X1+Abst,230-We(I)
  X1=X1+Teil
Next I
Text 10,290,"Das aritmetische Mittel ist :"+Miwe$
```

Listing 1a

daß die Eingabe an einer gewünschten Stelle auf dem Bildschirm erfolgt. Dafür muß man nur die Zeile bzw. Spalte eingeben:

INPUT @ (Zeile, Spalte);Var__Name

Obwohl dieser Befehl sehr leistungsfähig ist, ist er nicht für jede Eingabe genügend geeignet. Wir hatten weiter oben erklärt, daß diese Anweisung auf eine 'erlaubte Eingabe' wartet, um im Programm fortzufahren. Diesen Punkt werden wir jetzt genau betrachten. Ein großes Manko vom INPUT ist, daß das Komma als Trennzeichen erkannt und nur der Teil links vor dem Komma angenommen wird. Der Rest wird einfach nicht berücksichtigt. Möchte man eine Realzahl als String eingeben, so wird dieser sozusagen in einen Integer verwandelt.

INPUT Zahl$? 123,456 <RETURN>
123

Das erste Beispiel wird den Einsatz dieses Befehls anschaulicher machen (siehe Listing 1 für OMIKRON und 1a fur GFA). Es handelt sich um ein minimales Balkendiagramm-Programm, das keinen großen Anspruch erhebt. Jetzt zur Erläuterung: Erst wird die Menge aller Angaben verlangt und dadurch ein Feld dimensioniert. Die verschiedenen Eingaben erfolgen über den INPUT-Befehl, der sich innerhalb der FOR...NEXT-Schleife befindet. Die variable Summe enthält die Addition aller Eingaben und wird später zur Bildung des Mittelwertes benutzt. Der zweite Teil des Programms hat die Aufgabe, die verschiedenen Blöcke auf dem Bildschirm darzustellen. Zuletzt wird das arithmetische Mittel ausgedruckt.

Beide Interpreter benutzen für dieses Programm fast die gleichen Anweisungen. Unterschiede gibt es nur bei einigen Grafikbefehlen. Es gibt eine Erweiterung des INPUT-Befehls, die solche Einschränkungen behebt. LINE INPUT macht in der Regel nichts anderes als die INPUT-Anweisung. Werden Kommas mit eingegeben, so werden diese nicht als Trennzeichen interpretiert, sondern sie werden genauso wie ein anderes Zeichen aufgenommen. Die LINE-INPUT-Anweisung ermöglicht allerdings nur die Eingabe von Zeichenketten (Strings). Möchte man reine Zahlen als Ergebnis haben, so muß man diese erst als String eingeben und später entsprechend umwandeln.

LINE INPUT Zahl$? 123.456
Zahl = VAL(Zahl$)

Es gibt noch einen Befehl zur Eingabe von Zeichenketten, der in beiden Interpretern (GFA und OMIKRON) vorhanden ist. Es handelt sich auch hier mehr oder weniger um eine Erweiterung des schon beschriebenen Befehls INPUT.

Es gibt Fälle, in denen man die Eingabe einer Zeichenkette auf eine bestimmte Anzahl von Zeichen begrenzen will. Das ist zum Beispiel der Fall bei der Eingabe von Daten, die zu einer Datei gehören, oder bei der Eingabe eines Passwortes, das die Benutzung eines Programms erst ermöglicht. Dafür hat der BASIC-Interpreter den Befehl INPUT$, der eine Eingabe auf eine gewünschte Länge begrenzt. Hat man diese Grenze erreicht, dann setzt das Programm sofort die Abarbeitung fort. Die große Einschränkung dieser Anweisung ist, daß die Zeichen nicht unmittelbar nach der Eingabe dargestellt werden. Die Eingabe wird einer Variablen zugewiesen, die später durch den Befehl PRINT auf dem Bildschirm angezeigt wird oder an einen anderen Teil des Programmes weitergegeben werden kann. Die Syntax von INPUT$ weicht sehr von den anderen INPUT-Anweisungen ab:

Eingabe$ = INPUT$ (Länge)

Obwohl dieser Befehl gegenüber dem normalen INPUT eine Erweiterung darstellt, ist er wegen der obengenannten Einschränkung nur in wenigen Fällen brauchbar. Wir werden später sehen, daß sich dieser INPUT$ in gewisser Weise wie der Befehl INKEY$ verhält.

Das abgebildete kleine Programm „PASSWORD" hilft uns, den INPUT$-Befehl in der Praxis zu betrachten (siehe Listing 2 für OMIKRON und 2a für GFA).

Die Atari-Sonderzeichen und der INPUT-Befehl

Wie Sie vielleicht wissen, umfaßt der Zeichensatz von Atari 255 Zeichen. Darunter befinden sich alle Buchstaben und Zahlen, die man auch von anderen Computern kennt. Man kann

```
100 'Beispiel INPUTS in OMIKRON
110 CLS
120 PRINT "Bitte Passwort eingeben"
130 Passwort$= INPUTS(5)
140 Geheim$="AMIGA"
150 '
160 IF Geheim$=Passwort$ THEN
170   FORM_ALERT (1,"[1]["+Passwort$+"|Richtig mein
180 ELSE                              Freund][Gut]",Button)
190   FORM_ALERT (1,"[1][Falsch Du Schurke][Schlu"]",Button)
200   'wie w|re es hier mit  SYSTEM
210 ENDIF
```

Listing 2

```
' Beispiel INPUT$ in GFA
'
Print "Bitte geben sie ihrer Passwort ein"
Passwort$=Input$(5)
Geheim$="AMIGA"  ! groß geschrieben !!
'
If Geheim$=Passwort$ Then
  Alert 1,Passwort$+"|Richtig mein Freund",1,Passwort$,A
Else
  Alert 1,"Falsch Du Schurke",1,"Schluß",Dum
Endif
```

Listing 2a

```
100 ' Beispiel INPUT USING in OMIKRON
110 DIM Eingabe$(3)
120 CLS
130 BOX 20,20,600,300
140 TEXT STYLE =1
150 REPEAT
160    READ Feld_N$
170    TEXT 40,50+Unter,Feld_N$
180    Unter=Unter+46.5
190    I=I+1
200 UNTIL I>=4
210 DATA "Name","Vorname","Adresse","Telefon"
220 Zeile=2
230 FOR I=0 TO 3
240    READ Steuer$,L_Ge
250    INPUT @(Zeile,20);Eingabe$(I) USING Steuer$,Tasten
260    Zeile=Zeile+3                              code,L_Ge
270 NEXT I
280 DATA "a+-+ ",10,"a+-+ ",10,"a0+ +-",20,"0-/+/+ ",15
```

Listing 3

```
' Beispiel FORM INPUT in GFA
'
Dim Eingabe$(3)
Cls
Box 20,20,620,300
Deftext 1,0,0,13
Repeat
  Read Feld_n$
  Text 40,50+Unter,Feld_n$
  Unter=Unter+46.5
  Inc I
Until I=>4
Data Name,Vorname,Adresse,Telefon
Zeile=3
For I=0 To 3
  Read Laenge
  Print At(20,Zeile);
  Form Input Laenge,Eingabe$(I)
  Zeile=Zeile+3
Next I
Data 20,10,20,5
```

Listing 3a

sie problemlos über die Tastatur eingeben und auf dem Bildschirm oder Drucker darstellen. Der ST besitzt aber noch andere Zeichen, die nicht so leicht zu handhaben sind.

Im ersten Teil dieser Serie haben wir gezeigt, wie durch den Befehl TEXT alle Zeichen des Atari-Zeichensatzes darstellbar sind. Viele von diesen Zeichen kann man problemlos in eine Variable übergeben. Für die 'normalen' Zeichen wie Buchstaben, Zahlen oder Operanden-Zeichen reicht die Betätigung einer Taste. Andere lassen sich nur durch gleichzeitiges Drücken von mehreren Tasten darstellen. Das ist z. B. der Fall bei allen Buchstaben, die mit einem Akzent (französische Zeichen) versehen sind, oder es tritt auch beim Wurzel-Zeichen auf. Andere Zeichen lassen sich über den INPUT-Befehl überhaupt nicht darstellen. Das betrifft die ersten 30 Symbole des Atari-Zeichensatzes.

Das Editieren bei INPUT

Es kann passieren, daß man bei der Eingabe einer Zeichenkette oder Zahl irgendwelche Eingaben falsch macht oder daß man sich vertippt hat. Es wäre sehr ärgerlich, wegen so einer Lappalie den gesamten Text wieder neu eingeben zu müssen. Die Entwickler beider BASIC-Interpreter (GFA und OMIKRON) haben mit dieser Möglichkeit gerechnet und vielfältige Möglichkeiten der Editierung zugelassen. Löschen, Bewegen des Cursors nach links und rechts sowie schnelleres Einrücken des Cursors am Anfang bzw. Ende der Eingabe sind möglich.

Sonderwünsche gefällig?

Wir haben die verschiedenen Varianten der INPUT-Befehle gesehen. Jede Form ist für einen bestimmten Fall besser geeignet. Nach der Beschreibung der obengenannten Befehle ist uns klar, daß jeder sogenannte Standard für beide Interpreter vorhanden ist. Man kennt aber von anderen Rechnern, daß das, was als Standard bezeichnet wird, sehr eingeschränkt und die Abweichung sehr groß ist. Andererseits möchte jeder Entwickler mehr oder weniger seine persönlichen Merkmale im Produkt unterbringen. Wir werden jetzt die Abweichung von INPUT-Varianten, die es bei den beiden erwähnten Interpretern gibt, betrachten.

Eine Eingabe mit definierter Länge bei GFA

Bei der Beschreibung des INPUT$-Befehls haben wir gesagt, daß die eingegebenen Zeichen nicht unmittelbar dargestellt werden. Das ist mit Sicherheit ein Manko, das diese Anweisung nicht sehr hilfreich macht. Der Entwickler von GFA-BASIC hat die Einschränkung dieser Befehle erkannt und etwas neues geschaffen, das wesentlich leistungsstärker ist.

Die FORM-INPUT-Anweisung ermöglicht die Eingabe einer Zeichenkette mit einer vordefinierten Länge, wobei die Zeichen direkt auf dem Bildschirm angezeigt werden. Ist die Länge erreicht, wird das Programm nicht fortgesetzt, sondern der Cursor bleibt stehen und das Programm wartet, bis ein RETURN eingegeben wird. Die Syntax entspricht ungefähr der des einfachen INPUT:

FORM INPUT Laenge,Variablename$

Betrachten wir jetzt anhand eines Beispiels, wie dieser Befehl funktioniert. Stellen Sie sich vor, Sie möchten eine Kartei mit den Namen, Adressen und Telefonnummern all Ihrer Freunde einrichten. Wegen des Speicherplatzes möchten Sie die verschiedenen Eingaben auf eine bestimmte Anzahl von Zeichen begrenzen. So werden zum Beispiel für das Feld „Nachname" nur 20 Zeichen reserviert und für das Feld „Vorname" nur zehn. Durch den Befehl INPUT$ können Sie diese Angaben problemlos auf die gewünschte Länge begrenzen. Sie werden aber nicht sehen, ob Ihre Eingabe richtig ist oder nicht. Außerdem, wenn die Eingabe die definierte Länge erreicht hat, wird das Programm sofort fortfahren (siehe Listing 3a).

```
100 ' Beispiel INKEY in OMIKRON
110 CLIP 0,0,639,399'
120 ' clip 0,0,319,199    '  bei Farbe
130 CLS
140 REPEAT
150   A$= INKEY$
160   B$= RIGHT$(A$,1)
170   IF B$<>"" THEN GOSUB Aus
180 UNTIL B$="x"
190 END
200 '
210 '
220-Aus
230 A$= MID$(A$,2,1)
240 A= ASC(A$)
250 X=(A=75)*2 OR(A=77)*-2
260 Y=(A=72)*2 OR(A=80)*-2
270 IF X<>0 OR Y<>0 THEN
280   Xa=Xa+X
290   Ya=Ya+Y
300   CIRCLE Xa,Ya,15
310 ENDIF
320 RETURN
```

Listing 4

```
' Beispiel INKEY in GFA
'
Do
  A$=Inkey$
  If A$<>""
    Gosub Aus
  Endif
  Exit If A$="x"
Loop
'
Procedure Aus
  A=Asc(Right$(A$))
  X=(A=75)*2 Or (A=77)*-2
  Y=(A=72)*2 Or (A=80)*-2
  If X<>0 Or Y<>0
    Xa=Xa+X
    Ya=Ya+Y
    Circle Xa,Ya,15
  Endif
Return
```

Listing 4a

Mit der FORM INPUT von GFA sind all diese Einschränkungen behoben. Das kleine Programm FORM INPUT soll zeigen, wie man diesen Befehl benutzen kann. Erst wird ein Feld dimensioniert, das die Aufnahme von vier verschiedenen Zeichenketten ermöglicht ('DIM Eingabe$(3)' nicht vergessen). Die Dimensionierung fängt mit Null an. Dann wird eine Box gezeichnet und der Titel jedes Feldes wird innerhalb dieser Umrandung dargestellt. Das Programm holt sich aus den DATA-Zeilen die verschiedenen Namen der Felder. Das geschieht alles in einer REPEAT...UNTIL-Schleife. Ist die Bedingung erfüllt (I=>4), wird die Schleife verlassen und das Programm fährt fort.

Die zweite Struktur ermöglicht die Eingabe durch den Befehl FORM INPUT. Die Länge jeder Eingabe holt sich das Programm ebenfalls aus einer DATA-Anweisung. Mit PRINT AT (20,Zeile); wird der Cursor positioniert. Die Variable ZEILE wird bis zum Wert Drei inkrementiert, die den Abstand der verschiedenen Felder beinhaltet.

Eine Eingabe mit Format von OMIKRON

Die OMIKRON-Entwickler haben sich einen anderen Leckerbissen ausgedacht. Sie erinnern sich wahrscheinlich an die Beschreibung des Befehls PRINT USING in der ersten Folge dieser Serie (Juni-Heft). Dieser Befehl dient dazu, eine formatierte Ausgabe darzustellen. Dazu benutzt BASIC verschiedene Steuerzeichen, die zu diesem Zweck vordefiniert sind. Der Befehl INPUT USING, der Bestandteil des Sprachumfanges von OMIKRON-BASIC ist, arbeitet sehr ähnlich – nur daß es sich hier nicht um eine Ausgabe, sondern um eine Eingabe handelt.

Der Befehl ist außerordentlich leistungsfähig, obwohl er von der Syntax her um einiges komplizierter ist als der einfache INPUT. Er wurde in erster Linie für kommerzielle Anwendungen gedacht.

Um die genauere Syntax zu erklären, werden wir jetzt das gleiche Beispiel verwenden wie eben bei dem FORM-INPUT-Befehl von GFA. Wenn Sie das Programm genau betrachten, werden Sie feststellen, daß die Unterschiede sehr gering sind. In Zeile 150 wird der Befehl INPUT USING so gesteuert, daß er ungefähr das gleiche tut wie das FORM INPUT von GFA. Die Steuerzeichenkette holt sich den Interpreter aus der DATA-Anweisung (siehe Listing 3).

Wo bleibt mein Schlüssel?

Sowohl GFA wie OMIKRON übermitteln eine Tastatureingabe in anderer Form. Der INKEY$-Befehl liefert immer die Information der zuletzt gedrückten Taste. Bei OMIKRON ist diese Information um einiges ausführlicher. Bei diesem BASIC-Interpreter wird immer in einem Vier-Byte-Paket der Zustand der SHIFT-Taste, der SCAN-Code und der ASCII-Code der gedrückten Taste geliefert.

Bei GFA wird prinzipiell nur der ASCII-Wert geliefert. Wird eine Sondertaste betätigt, so wird ein Zwei-Zeichen-String übergeben, wobei das zweite Zeichen den Code für die Sonder-Taste beinhaltet.

Die INKEY$-Anweisung unterbricht das Programm nicht. Wird keine Taste gedrückt, so wird ein leerer String zurückgeliefert. Genau deswegen wird dieser Befehl meist innerhalb einer Schleife verwendet. Der Befehl ist für die Eingabe von längeren Zeichenketten nicht sehr geeignet, aber für die Abfrage bestimmter Tasten fast ideal. Die Syntax ist sehr unkompliziert und für beide Interpreter-Versionen gleich:

```
Eingabe$=INKEY$
oder
REPEAT
  :
  EXIT IF Eingabe$="X"
  :
UNTIL...
```

Das nächste Beispiel soll die Funktion von INKEY$ in der Praxis zeigen (siehe Listing 4 für OMIKRON und 4a für GFA).

Der INKEY$-Befehl wird innerhalb einer DO...LOOP-Schleife (bei OMIKRON-BASIC in einem REPEAT UNTIL) benutzt und dadurch ständig abgefragt. Wurde keine Taste gedrückt, so bleibt das Programm in der Schleife. Beim Drücken einer Cursor-Taste wird das Programm in die Procedure 'Aus' umgeleitet. Dort wird die Taste identifiziert. Das geschieht durch eine logische Verknüpfung und eine Vergleichsoperation. Bei so einer Vergleichsoperation in BASIC erhält man bei einer Wahr-Aussage eine −1 (TRUE) als Ergebnis zurück und bei einer falschen Aussage eine 0 (FALSE). Betrachten wir die folgenden Beispiele:

```
A = 3
B = 4
C = 3
X = A = B
Z = A = C
Print X,C
```

Auf dem Bildschirm:
0 −1

```
A$ = "Meier"
B$ = "Meyer"
C$ = "Meier"
X  = A$ = B$
Z  = A$ = C$
Print X,Z
```

Auf dem Bildschirm:
0 −1

Das bedeutet, daß die erste Aussage nicht richtig ist (falsch), und daß es sich bei der zweiten um eine wahre Aussage handelt.

Das gleiche können wir mit Zeichenketten (Strings) wiederholen:
In unserem Beispiel wird der folgende Vergleich gemacht:

X= (A=75)★2 OR (A=77)★ −2

Wird zum Beispiel die Taste 'Pfeil nach links' gedrückt, so ist die Aussage X = (A=75) wahr und X wird eine −1 zugewiesen. Danach folgt die Multiplikation mit 2, so daß der Endwert von X eine −2 ist. Wurde nicht die Taste 'Pfeil nach links' gedrückt, so wird zuerst überprüft, ob der 'Pfeil nach rechts' betätigt wurde. Ist das auch nicht der Fall, dann werden die zwei anderen Richtungen kontrolliert. Wir benutzen diesen Vergleich, um einen Inkrementwert (XA = XA+X bzw. YA= YA+Y) zu gewinnen, der später für das Zeichnen eines Kreises (CIRCLE XA,YA) auf verschiedenen Stellen des Bildschirms verwendet wird.

Damit ist die zweite Folge beendet. Wir werden uns im nächsten Heft mit anderen Kapiteln der BASIC-Programmierung beschäftigen.

(HS/MM)

Die Neuen und die Erfolgreichen aus dem Heim-Verlag

ATARI ST
Prof. Dr. B. Bellow/K. Reimann
GfA-BASIC Programm-Sammlung

B-407 Buch 49,– DM
D-427 Diskette 49,– DM

Anhand von über 130 Programmbeispielen und Erläuterungen lernen und trainieren Sie das Programmieren in BASIC. Von einfachen, aber grundlegenden Beisp. bis zur ausgereiften Anwendung findet der ATARI ST-Besitzer Beispiel-Programme, die den perfekten Einstieg in die Programmiersprache des GfA-BASIC's leicht machen. Alle Beispiele werden so dargestellt, daß das Verständnis für die Programmstruktur gefördert wird, aber auch die Details der GfA-BASIC-Version deutlich werden. Die Progr. sind gut dokumentiert, wobei die hervorragenden Möglichkeiten des GfA-Basic zur strukturierten Programmierung genutzt werden. **Aus dem Inhalt:** ● Eigenschaften des GfA-BASIC ● Dateiverwalt. u. Ordner in GfA-BASIC ● Schleifentechnik ● Felder ● Unterprogramme ● Menü-Steuerung ● Window Technik ● Zufallszahlen ● Seq.- u. RANDOM-Dateien ● Textverarbeitung ● Sortierprogramme ● Fakturiersysteme ● Grafik ● Spiel uvm. 320 Seiten mit 131 Programmen in GfA-BASIC

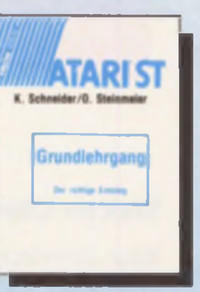


B-408 Buch 49,– DM
D-428 Diskette 39,– DM

Wenn Sie das Software-Paket **VIP-Professional** kaufen wollen oder schon besitzen, dann weiht Sie dieses Buch schnell und umfassend in die Geheimnisse dieses Profiprogrammes ein.
VIP-Professional besteht aus den drei Funktionsbereichen

- DATENBANK
- KALKULATION
- GRAFIK

mit denen wichtige und vielfältige Aufgaben hervorragend gelöst werden.
Das Buch enthält komplette Musterlösungen für die Gewinn- u. Verlustrechnung und Fakturierung. Anhand dieser Beispiele wird gezeigt, wie Sie alles herausholen, was in VIP-Professional steckt.
Mit diesem Buch können Sie **VIP-Professional** richtig einsetzen und seine Möglichkeiten voll ausschöpfen.



B-400 Buch 49,– DM
D-420 Diskette 39,– DM

Das Buch für den richtigen Einstieg mit dem ATARI ST. Auf rund 330 Seiten wird der Leser leicht verständlich in die Bedienung des Rechners eingeführt.
Einige der Themen:

Die Hardware des ATARI ST:
- Aufstellung und Wartung des Computers
- Überblick über die Systemkomponenten
- Das „Innenleben" des Computers

Die Software des ATARI ST:
- Die Bedienung des GEM-Desktop
- Arbeiten mit Maus, Fenstern und Icons
- Einführung in die Sprachen LOGO und BASIC
- Programmsammlung mit vielen interessanten Beispielen

Diese zweite Auflage des Grundlehrgangs berücksichtigt alle neuen Rechnermodelle der ATARI ST-Familie.



B-402 Buch 49,– DM
D-422 Diskette 39,– DM

Der ideale Einstieg in die Programmierung mit LOGO. Ein Buch für alle, die LOGO schnell verstehen und perfekt erlernen möchten. Ein Buch wie es sein muß, leicht zu verstehen und interessant geschrieben.
Einige der Themen:
- **Variablen- und Listenverarbeitung**
- **Packages und Properties**
- **Die Turtlegrafik**
- **Viele ausführlich erklärte Beispielprogramme zu den Themen**
  - Arbeiten mit relativen und sequentiellen Dateien
  - Anwendungen aus Mathematik
  - Hobby- und Spielprogramme u.v.a.m.

Zum Buch gibt es die Programmdiskette mit sämtlichen Beispielprogrammen. Ca. 350 Seiten mit vielen Abbildungen.



B-405 Buch 49,– DM
D-425 Diskette 39,– DM

Zum neuen Basic Interpreter, ein Buch, das mit gezielten Beispielen verständlich den Einstieg in das Basic der Superlative ermöglicht. Ein Muß für jeden Besitzer dieses Interpreters.
Einige der Themen:
- Der Umgang mit dem Editor
- Ausführliche, mit Beisp. versehene Befehlsübersicht
- Die fantastischen Grafikmöglichkeiten (Windows, Sprites, Alertbox, Pull-down Menüs
- Strukturierte Programmierung – auch in Basic möglich!
- Dateiverwaltung unter GfA-Basic
- GEM Handhabung in Basic
- Vergleich mit anderen Basic-Interpretern – Hilft bei der Kaufentscheidung
- Zahlreiche Übungs- und Anwenderbeispiele
- Mathematik und hohe Genauigkeit



B-409 Buch 49,– DM
D-429 Diskette 39,– DM

Ein Standardwerk für den ATARI ST – ideal für Schüler und Studenten! Anhand von zahlreichen Beispielen lernen Sie die Lösung von mathematischen Problemen mit dem ATARI ST kennen. Die zahlreichen BASIC-Programme sind praxisnah ausgewählt worden und können in Schule, Studium und Beruf eingesetzt werden.

Einige der Themen:
- Integral- und Differentialrechnung
- Kurvendiskussion
- Statistik und Wahrscheinlichkeitsrechnung
- Matrizenrechnung
- Radioaktivität
- Relativitätstheorie

u.v.a.m.



B-406 Buch 49,– DM
D-426 Diskette 39,– DM

C ist die zweite „Muttersprache" des Atari ST: schnell, komfortabel, kompakt im Code.
„C auf dem Atari ST" ist für Anwender geeignet, die Erfahrungen mit anderen Programmiersprachen gemacht haben. Das Buch behandelt den vollen Sprachumfang von Standard-C und verweist auf BASIC und Pascal.
Besonderer Wert wird auf die Anschaulichkeit und Genauigkeit der Darstellung gelegt. Alle Programmbeispiele sind getestet und direkt in den Text übernommen.
Aus dem Inhalt: C-Compiler für den Atari: Digital Research, Lattice, Megamax ● der Editor ● Bedienung des Compilers ● Grundlegende Elemente eines C-Programms ● Variablentypen ● Felder und Vektoren ● Ausdrücke ● Zeiger ● Speicherklassen ● Bitfelder ● Varianten ● Aufzählungen ● Dateien ● Diskettenhandling ● Einbindung von Assemblerprogrammen ● Bildschirmgrafik in C ● Fehler in den C-Compilern ● Tools u.a.



B-404 Buch 49,– DM
D-424 Diskette 39,– DM

Ein Grundwerk.
Wer sich mit der Programmierung der GEM-Funktionen vertraut machen will braucht dieses Buch! Es beginnt mit einer Erläuterung des GEM-Aufbaus und führt anhand von Programmbeispielen zum **leichten Verständnis aller auf dem Atari verfügbaren GEM-Funktionen.** Die übersichtliche Gliederung ermöglicht auch ein schnelles Nachschlagen der Funktionen.
Einige der Themen:
● Was ist **GEM** ● Die GEM-Bestandteile **VDI** und **AES** ● Die **GEM-Implementation** auf dem Atari ST ● Aufruf der GEM-Funktionen aus **BASIC, C,** und **ASSEMBLER** ● Sonstige **Programmiersprachen** und GEM ● Die Programmierung der **VDI**-Funktionen ● Die **AES**-Bibliothek und die Programmierung ihrer Funktionen ● Aufbau eines **Objektbaumes** ● Was ist eine **Resource**-Datei? ● Viele erläuterte **Beispielprogramme** in BASIC, C und ASSEMBLER.

ACHTUNG: Zu den Büchern gibt es die Programmdiskette mit allen Programmbeispielen – Diskette laden und los geht's...

Für Bestellungen verwenden Sie am besten die in der ST-Zeitschrift vorhandene BUCH- UND SOFTWARE-BESTELL-KARTE.
Bücher und Programm-Disketten aus dem Heim-Verlag erhalten Sie auch bei Ihrem ATARI-Fachhändler oder im Buchhandel.

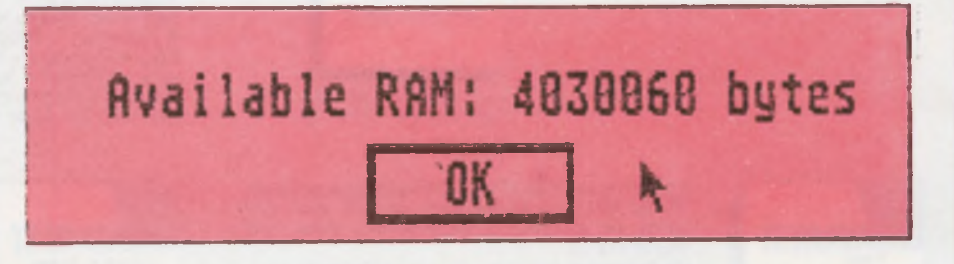


Telefon 0 61 51 / 5 60 57
6100 Darmstadt-Eberstadt · Heidelberger Landstr. 194

*alle Preise sind unverbindliche Preisempfehlungen

ST-Tuning: Speichererweiterungen für den ST

Während Atari die neuen Mega-Modelle mit 2 und 4 Megabyte Speicherkapazität ausliefert, bieten gleich mehrere Hersteller solche Speichererweiterungen für die „alten" STs an. Wir testeten vier unterschiedliche Lösungen, die den ST in einen Speicherriesen verwandeln.



Wozu soviel Speicher?

Viele Anwender mögen sich fragen, wozu ein RAM-Speicher mit 1, 2 oder sogar 4 MByte Kapazität überhaupt notwendig sei. Doch die Zeiten, in denen ein Rechner mit 48 oder 64 Kilobyte seinen Benutzer in größtes Erstaunen versetzte, sind längst vorbei. Gerade beim Atari ST mit seiner grafischen Benutzeroberfläche (GEM) und der Möglichkeit, Accessories (Hilfsprogramme) ständig im Speicher parat zu haben, muß man umdenken – zumindest, was die Programmlänge und den damit verbundenen Speicherplatzbedarf angeht. Bestimmte Anwendungen machen einen Speicher von mindestens 1 MByte sogar unbedingt erforderlich: Zum Beispiel computerunterstütztes Zeichnen (CAD), Desktop Publishing (bildschirmorientiertes Satzsystem), bewegte Grafiken (Animation), digitale Ton- und Bildverarbeitung (Sound Sampling, Bilddigitalisierung) und anderes mehr. Natürlich werden auch diejenigen Programmierer, die ein komplettes Programmiersystem mit Compiler, Linker usw. auf einer großen RAM-Disk betreiben, einen entsprechend großen Speicher zu schätzen wissen. Schließlich lassen sich hochauflösende Grafiken auf dem ATARI-Laserdrucker nur mit mindestens einem 2 MByte-ST ausdrucken, da der Drucker nicht über einen eigenen Speicher verfügt. Wie Sie sehen, gibt es also zahlreiche Anwendungen, die ein bequemes und sinnvolles Arbeiten erst bei einem Arbeitsspeicher ermöglichen, der größer als 1 MByte ist.

Welche Lösung bietet sich an?

Eine Erweiterung des Speichers auf 2 oder sogar 4 MByte im Selbstbau, ähnlich unserer Bauanleitung in Heft 1/86, in dem wir die Aufrüstung von 512 KByte auf 1 MByte ausführlich beschrieben hatten, erscheint uns hier nur für den sehr versierten Elektronik-Bastler geeignet. Zudem sind die Preise der Megabit-RAMs für den „Normalverbraucher" bei kleinen Stückzahlen zur Zeit noch sehr hoch, so daß sich ein Selbsteinbau schon aus dieser Sicht kaum lohnt – ganz zu schweigen vom Risiko einer eventuellen Rechnerreparatur nach einem gescheiterten Umbauversuch. Beschränken wir uns also auf die käuflichen Speichererweiterungen.

Grundsätzliches zur Speicherverwaltung im Atari ST

Die Atari-Modelle 520 ST, 260 ST sowie der 520 STM verfügen standardmäßig über 512 KByte Speicherkapazität. Die Modelle 520 ST+ und der 1040 STF besitzen jeweils einen RAM-Speicher von 1 MByte. Heute werden nur noch der 520 STM, der 1040 STF sowie natürlich die neuen Mega-Modelle ST2 und ST4 produziert. Dieser Systemspeicher von 512 KByte bzw. 1 MByte wird von einem der vier Atari-eigenen Custom-Chips, der sogenannten **MMU** (**M**emory **M**anagement **U**nit oder Speicherverwaltungseinheit) verwaltet. Die angeschlossenen **RAM**-Bausteine (**R**andom **A**ccess **M**emory, Speicher mit beliebigem Zugriff) sind dabei in zwei Bänke mit je 16 RAM-Chips organisiert. Es ist möglich, beide Bänke mit unterschiedlichen RAM-Bausteinen zu bestücken, doch muß jede Bank für sich mit Chips derselben Art ausgestattet sein. Folglich ist es nicht möglich, **eine** Bank mit z. B. acht 256 KBit-Chips und weiteren acht 1024 KBit (1 MBit)-Bausteinen zu bestücken. Welche Bausteine in einer RAM-Bank verwendet werden, erkennt der Rechner beim Kaltstart (Systemstart) automatisch und programmiert ein Register der MMU ($FF8001) entsprechend. Die MMU kann bei voller Bestückung mit den neuen Mega-Chips maximal 4 MByte RAM-Speicher verwalten.

Beim guten „alten" 520 ST wurde eine RAM-Bank mit den 256 KBit-Chips bestückt, so daß dieser Rechner einen nutzbaren Speicherraum von 512 K-Byte erhielt. Diese Kapazität erwies sich bald als nicht ausreichend, so daß bei den Nachfolgemodellen, dem 520 ST+ und dem 1040 STF, auch die zweite RAM-Bank mit sechzehn 256 KBit-Chips bestückt wurde. Dadurch verdoppelte sich der Speicherraum auf 1 MByte. Eine weitere Vergrößerung des Speichers mit 256 KBit-Chips ist, bedingt durch die MMU, die nur zwei RAM-Bänke verwalten kann, nicht vorgesehen.

Lange Wartezeiten

Wie erwähnt, prüft der Rechner – besser gesagt, das TOS – beim Systemstart, wieviel RAM-Speicher vorhanden ist. Dabei wird jede Speicherzelle der RAMs geprüft, was natürlich eine gewisse Zeit in Anspruch nimmt. Man muß sich also bei einem 4 MByte-Rechner auf eine deutlich längere Bootphase als bei einem normalen 260 ST mit 512 KByte Speicher einstellen. Doch nicht nur das Booten (Einschaltphase) dauert länger, sondern auch das Laden von Programmen – wobei natürlich nicht die Übertragungsgeschwindigkeit zwischen Floppy und Rechner gemeint ist. Vielmehr wird der gesamte Speicher vom Ende des eingeladenen Programms bis zur Speichergrenze mit einem Bytemuster gefüllt. Warum das TOS jedesmal den freien Speicher füllt, ist leider nirgends zu erfahren. Aber diese Tatsache hat nichts mit den getesteten Speichererweiterungen zu tun, sondern ist eine „normale" Eigenschaft des Betriebssystems. Zum Glück ist diese störende Eigenart im TOS der neuen Mega-STs verschwunden.

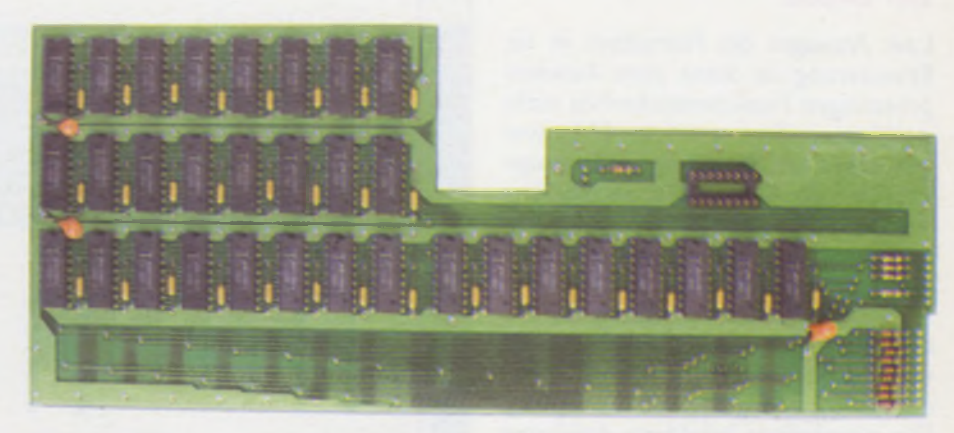


Bild 1: 4 MByte Erweiterung von Rhothron für den 1040 STF

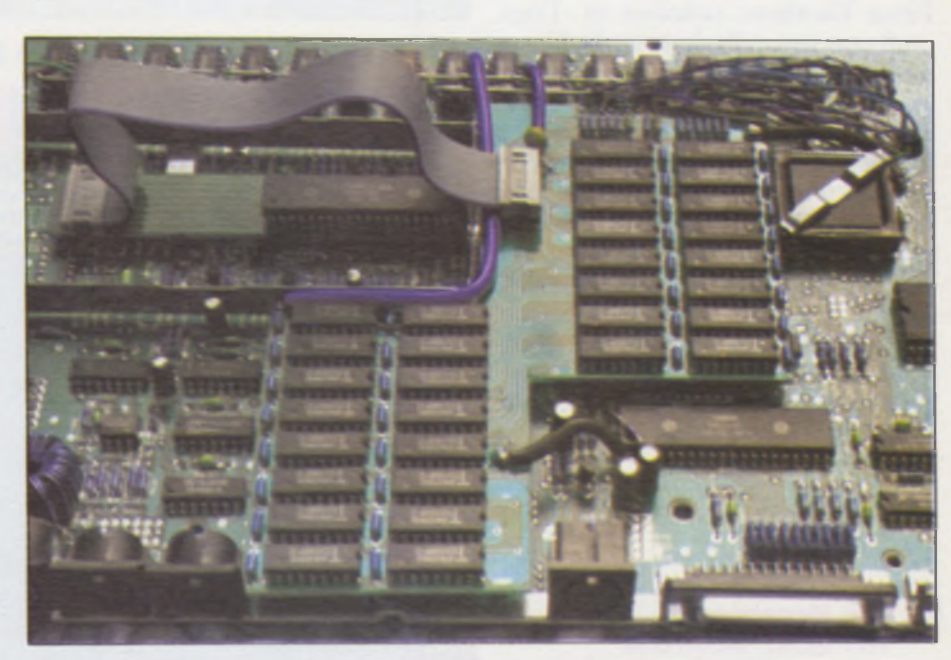


Bild 2: 4 MByte Erweiterung von Rhothron für den 260 ST

1.) 2 und 4 MByte-Erweiterung von Rhothron

Die Firma Rhothron bietet eine 2/4 MByte-Erweiterung für alle ST-Modelle außer den 520 STM an. Dabei unterscheiden sich das Platinenlayout der Erweiterung für den 1040 STF (siehe Bild 1) und das für die Modelle 520 ST, 260 ST und 520 ST+ (siehe Bild 2). Die zuletzt genannte Version könnte aus elektronischer Sicht auch am 520 STM betrieben werden, es gibt jedoch Platzprobleme, da der HF-Modulator des 520 STM einen Teil des von der Speicherweiterung benötigten Platzes belegt. Abhilfe könnte der Einbau des Rechners in ein anderes Gehäuse oder die Auslagerung der Speichererweiterung schaffen. Die Platine dieser Erweiterung ist für die 2 bzw. 4 MByte-Version identisch, sie unterscheidet sich nur durch die Zahl der RAM-Bausteine. Die 2 MByte-Version ist mit sechzehn, die 4 MByte-Erweiterung mit 32 CMOS-RAMs bestückt. Verwendung finden die modernen Megabit-Chips. Da die zweite RAM-Bank der 2 MByte-Erweiterung voll mit Sockeln bestückt ist, können die fehlenden sechzehn Chips jederzeit nachgerüstet werden, so daß der Rechner nachträglich auf volle 4 MByte erweitert werden kann. Durch den Einsatz der Mega-Chips wird die Speichererweiterung vollständig vom Betriebssystem unterstützt, d. h. für den Betrieb ist keine Bootsoftware notwendig. Ein weiterer Vorteil ist, daß auch die 4 MByte-Erweiterung in jedes normale Atari-Gehäuse paßt (mit Ausnahme des 520 STM).

1 MByte + 2 MByte = 2,5 MByte!

Die 2 bzw. 4 MByte-Erweiterung stellt keinen komplett zusätzlichen Speicherplatz dar, sondern ist vielmehr ein Ersatz der im Rechner bereits vorhandenen Speicher-Chips. Das bedeutet, daß ein Atari ST, der bisher mit 2x512 KByte (1 MByte) bestückt war, nach der Aufrüstung mit der 2 MByte-Erweiterung **nicht** 3 MByte, sondern „nur" 2,5 MByte freien Speicher besitzt. Eine der vorhandenen RAM-Bänke muß also der Erweiterung weichen. Sinngemäß gilt das gleiche für die 4 MByte-Erweiterung, hier ergibt sich bei allen Modellen ein frei verfügbarer Speicher von 4 MByte. Die bereits im Rechner befindlichen RAMs müssen alle deaktiviert werden.

Der Einbau

Laut Aussagen des Herstellers ist die Erweiterung im Sinne einer hundertprozentigen Funktionssicherheit nicht steckbar, sondern es müssen 17 Leitungen sowie die Spannungsversorgungskabel auf der Rechnerplatine angelötet werden. Außerdem müssen einige Leiterbahnen der Rechnerplatine durchgetrennt werden. Ein mitgelieferter Zwischensockel, der zwischen den Videoshifter und dessen Sockel gesteckt wird, stellt über ein Flachbandkabel die restlichen Signale zur Verfügung. Die mitgelieferte bebilderte Anleitung umfaßt 15 Seiten (DIN A5). In ihr wird der Einbau der 2/4 MByte-Speichererweiterung für alle ST-Modelle knapp, aber gut verständlich beschrieben, so daß der geübte Bastler die Erweiterung auch selbst einbauen kann, obwohl im Preis der Einbau durch die Firma Rhothron enthalten ist. Dazu muß man seinen Rechner an die Firma schicken, man bekommt ihn dann aufgerüstet und mit sechs Monaten Garantie auf die Speicherchips zurück. Im Lieferumfang ist eine Diskette mit diversen Programmen enthalten. Neben einer RESET-festen RAM-Disk wird ein Programm mitgeliefert, das den frei verfügbaren Speicherbereich ausgibt und ihn fortlaufend durch Schreib- und Lese-Operationen überprüft. Die 2 MByte-Aufrüstung kostet einschließlich Einbau 898,– DM, die 4 MByte-Aufrüstung 1598,– DM.

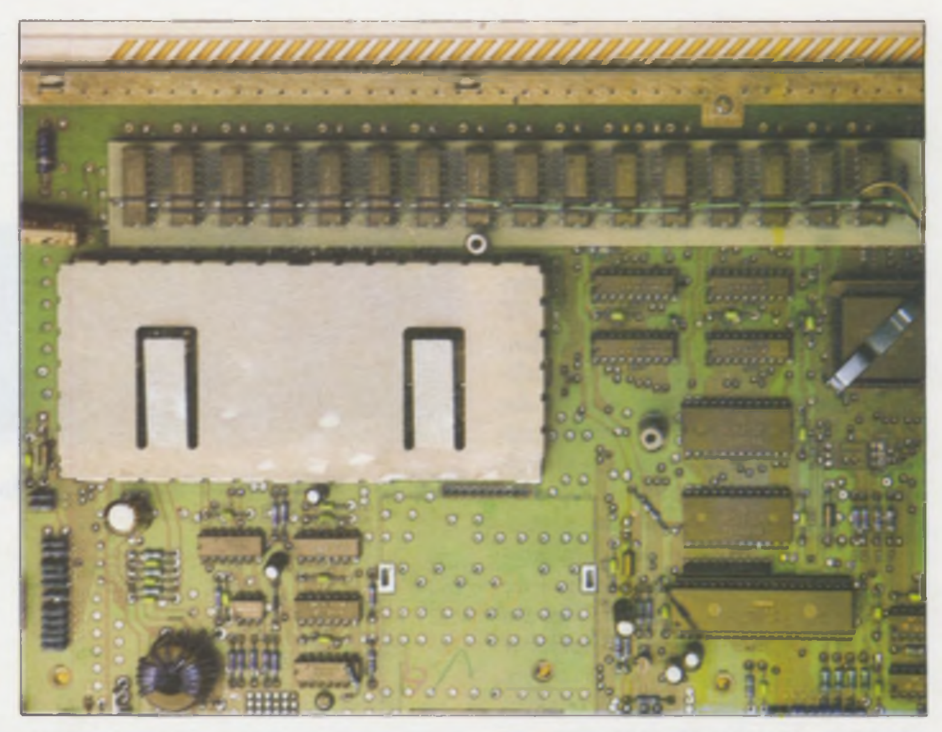


Bild 3: Das Megaboard mit 4 MByte RAMs

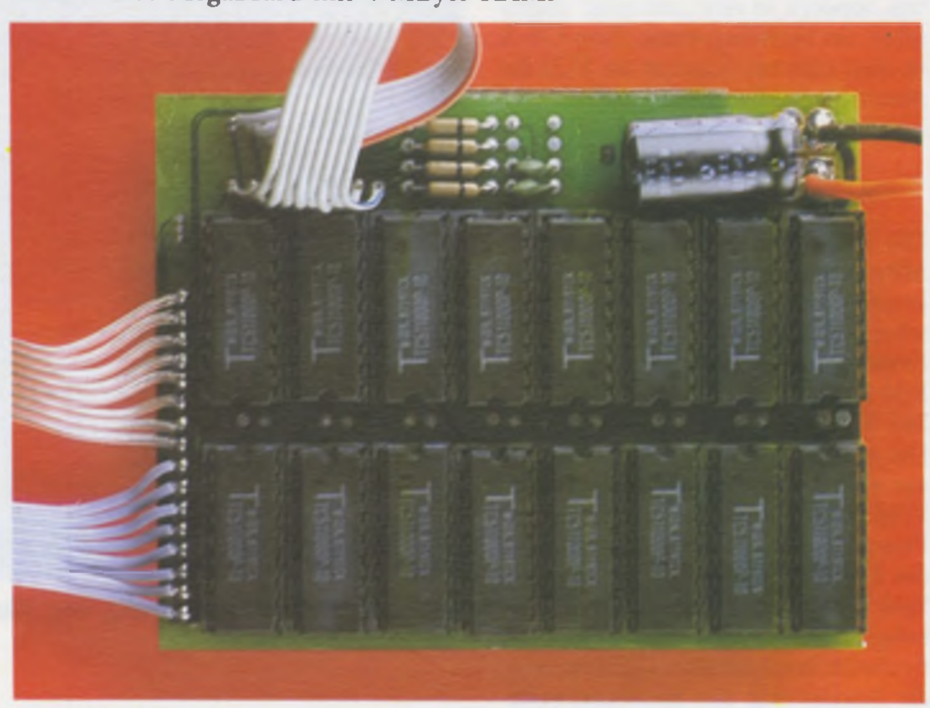


Bild 4: Steckbar 2 MByte Erweiterung von Weide-Elektronik

2.) Das Megaboard von ECKL electronic

Die Firma ECKL electronic bietet ebenfalls eine 2 bzw. 4 MByte-Speichererweiterung für **alle** ST-Modelle an. Auf einer schmalen Platine (siehe Bild 3) finden 16 1-Megabit-Chips Platz, so daß sich eine Kapazität von 2 MByte ergibt. Bei der 4 MByte-Version werden einfach, und dies kann auch zu einem späteren Zeitpunkt erfolgen, weitere 16 RAMs „huckepack" auf die vorhanden Mega-Chips aufgelötet. Dadurch ergibt sich eine sehr kompakte Lösung, die in alle ST-Rechner eingebaut werden kann. Beim Einbau wird eine Bank der bereits im Rechner befindlichen RAMs entfernt und stattdessen die Platine mit den Megabit-Chips eingelötet. Bedingt durch einen anderen Abstand der RAM-Bausteine im 1040 STF zu den restlichen ST Modellen wird für diesen Rechner eine andere Platine benötigt, die aber im Prinzip mit der auf Bild 3 abgebildeten Platine für den 260 ST (520 ST, 520 ST+ und 520 STM) identisch ist. Ferner werden für den 1040 STF bei einer Aufrüstung auf 4 MByte 2 Platinen 2 MByte benötigt, da bei diesem Rechner die zwei RAM-Bänke nebeneinander liegen. Aufgrund dieser Tatsache ergeben sich auch unterschiedliche Speicherkapazitäten nach dem Einbau der 2 MByte-Erweiterung beim 1040 STF im Gegensatz zum 520 ST+, bei dem ja die zweite RAM-Bank huckepack, also über die erste gelötet ist. Folgende Tabelle gibt eine Übersicht:

Rechner	verfügbarer Speicher nach Einbau der 2 MByte Erweiterung	4 MByte Erweiterung
1040 STF	2,5 MByte	4 MByte
520 ST+ 260 ST 520 STM	2 MByte	4 MByte

Das Megaboard ist abschaltbar

Das Megaboard wird nur von diversen Atari-Fachhändlern bzw. direkt vom Hersteller eingebaut, da mindestens 16 der im Rechner befindlichen RAMs ausgelötet werden müssen. Zwar muß man ein paar Tage auf sein Gerät verzichten, erhält dafür aber einen funktionstüchtigen Rechner mit sechs Monaten Garantie auf den neuen Speicher. Zum Einbau gehört ein kleiner Schalter, der in die Gehäuserückseite des Rechners eingebaut wird. Mit diesem Schalter, der übrigens nur beim Megaboard vorhanden ist, können Sie dann wahlweise zwischen 512 KByte und 2 MByte bei der 2 MByte-Version, bzw. zwischen einem und 4 MByte bei der 4 MByte-Version umschalten. Damit ist sichergestellt, daß auch bereits vorhandene Software weiterhin eingesetzt werden kann. Das Umschalten kann aber nur in ausgeschaltetem Zustand erfolgen, da der Rechner sonst abstürzt. Auch das Megaboard benötigt kein externes Netzteil und wird vom System automatisch erkannt, so daß keine Software zum Betrieb notwendig ist. Das Megaboard kostet inklusive Einbau in der 2 MByte-Version 998,– DM, die 4 MByte-Version für den 260 ST, 520 ST bzw. 520 STM kostet 1848,– DM und die gleiche Version für den 1040 STF 1995,– DM.

3.) Steckbare 2 MByte-Erweiterung von Weide-Elektronik

2 MByte im Selbstbau

Weide-Elektronik bietet nach altbewährtem Rezept eine 2 MByte-Speichererweiterung an, die Sie ohne Löten in den Rechner einstecken können. Uns stand die 2 MByte-Erweiterung für den 260 ST (520 ST und 520 ST+) zur Verfügung (Bild 4), die bestückt mit 16 1-MBit-Chips auf die Rechnerplatine aufgeklebt wird, an die Stelle, an der damals der HF-Modulator geplant war. Für die Rechnermodelle 1040 STF und den 520 STM existiert nach Angaben der Firma eine weitere Platine, die ebenfalls durch bloßes Einstecken installiert werden kann. Die benötigten Signale werden durch einen Zwischensockel vom Videoshifter und durch eine kleine Adapterplatine, die auf die MMU gesteckt wird, abgenommen. Zur Spannungsversorgung dienen zwei Krokodilklemmen, mit denen Masse und +5 Volt an der Rechnerplatine abgegriffen werden. Die 15seitige bebilderte Einbauanleitung (DIN A5) beschreibt den relativ einfachen Einbau für alle ST-Rechner recht gut.

Erreichte Speicherkapazität

Zum Betrieb der Erweiterung muß eine der beiden RAM-Bänke deaktiviert werden (ähnlich wie bei dem System von Rhothron), so daß Ihnen nach dem Einbau der Erweiterung immer 2,5 MByte freier Speicher zur Verfügung steht – egal, ob Ihr Rechner zuvor 512 KByte oder 1 MByte RAM-Speicher hatte. Auch diese Erweiterung benötigt zum Betrieb keine zusätzliche Software, da die MMU beim Kaltstart die Erweiterung erkennt.

4 MByte in Vorbereitung

Nach Angaben von Weide-Elektronik ist eine 4 MByte-Erweiterung für den Herbst geplant. Auch diese Erweiterung wird dann allein durch ein Stecksystem installiert werden. Wahrscheinlich wird diese neue Platine die bisherige 2 MByte-Erweiterung ablösen und dann je nach Bestückung als 2 bzw. 4 MByte-Erweiterung Verwendung finden.

4.) 1 und 2 MByte-Erweiterung mit 256-KBit-Chips

Bei den bisher genannten Systemen erfolgte die Erweiterung des ST-Hauptspeichers mit den Megabit-Chips. Die nun beschriebene Speichererweiterung von G. Dantrimont Systementwicklung benutzt die 256 KBit-Chips, wie sie auch in den bisherigen Atari-Rechnern benutzt werden. Dadurch bedingt sind auf der Erweiterungsplatine nicht nur die RAM-Bausteine vorhanden, sondern zusätzlich eine Steuerelektronik, die der MMU gewissermaßen die Megabit-Chips simuliert. Durch diesen „Kunstgriff" ergibt sich nicht nur ein günstiger Preis, sondern es können auch alle im Rechner vorhandenen RAMs weiterbenutzt werden. Um dies zu verdeutlichen, sind in der untenstehenden Tabelle die Speicherkapazitäten zu den drei lieferbaren Erweiterungen mit Preisen angegeben.

Da ein Speicherausbau über ein MByte eigentlich nur mit Megabit-Chips gedacht war (siehe oben), kennt das GEM auch nur bestimmte Speicher-Ausbaustufen, welche auch bei der Verwendung von Megabit-Chips erreicht werden könnten. Der Rechner erkennt daher beim Systemstart nur die folgenden Ausbaustufen: 0,5 MByte; 1 MByte; 2 MByte; 2,5 MByte und 4 MByte. Die in der Tabelle aufgeführten Zwischenstufen (mit einem Stern gekennzeichnet) müssen dem Rechner erst durch das mitgelieferte RAMTOP.PRG-Programm mitgeteilt werden. Befindet sich dieses Programm in einem AUTO-Ordner, so wird beim Booten dem Rechner der tatsächliche Speicherraum mitgeteilt.

Grund-Speicher	verfügbarer Speicher nach Einbau der 1 MB Karte	1,5 MB Karte	2 MB Karte
1 MByte 1040 STF 520 ST+	2 MByte	2,5 MByte	3 MByte*
512 KByte 260 ST 520 ST 520 STM	1,5 MByte*	2 MByte	2,5 MByte
Preis:	unter DM 600,–	unter DM 800,–	unter DM 1000,–

* = zum Betrieb Software notwendig (siehe Text)





BARBARIAN

"Liebe Deine Feinde, aber sei schneller als sie!" (Lao Tse, chinesischer Philosoph)

In Thelston lebten einst die Zwillingsbrüder Thoron und Necron. Ihr Lehrer war ein alter Druide, der sehr schnell die unterschiedlichen Neigungen der Brüder erkannte. Thoron war tapfer und gütig, Necron hingegen skrupellos und gerissen. Beide entwickelten große magische Fähigkeiten. Am Ende ihrer Ausbildung erhielt Thoron ein glänzendes Schwert, Necron einen wundervollen Bogen. Der Druide ermahnte Necron, sich niemals gegen seinen Bruder zu wenden.

Die Brüder wurden erwachsen und gingen ihrer Wege. Thoron wurde ein bekannnter Jäger und durchstreifte das Land nach Abenteuern. Necron verschwand. Jahre später trafen sich die Brüder in einer Stadt, die von Necron ausgebeutet und tyrannisiert wurde. Necron fiel im Kampf, aber die Mächte des Bösen retteten seine dunkle Seele. So wurde er der Herrscher der Unterwelt von Durgan und begann, mit seinen finsteren Gestalten das Land zu geißeln.

Indessen wurde Thoron ein Sohn geboren: Hegor. Bereits in jungen Jahren wurde er von seinem Vater im Kampf mit Schwert und Bogen ausgebildet. Tapfer, klug und abenteuerlustig war der junge Hegor, als er auszog, um der berühmteste Drachentöter aller Zeiten zu werden. Er durchstreifte das Land und war bald überall als tapferer Kämpfer bekannt. Necrons dunkle Horden hingegen brachten den König immer mehr in Bedrängnis. Der Drache Vulcuran tötete Thoron; schließlich wußte der Herrscher keinen anderen Ausweg mehr, als demjenigen, der der Zerstörung Einhalt böte und die bösen Mächte bezwänge, die Krone des Reiches und all seine Reichtümer zu versprechen. Hegor machte sich auf, den Tod seines Vater zu rächen. Ein weites Sumpfgebiet schützt den Eingang der Unterwelt von Durgan. Hegor zieht sein Schwert und schreitet voran...

Story, Szenario und Handlung des neuen Spiels von Psygnosis sind faszinierend. Wer einmal angefangen hat, sich mit Hegor durch die Unterwelt zu kämpfen, wird sehr schnell vom "Barbarian-Fieber" gepackt. Nicht immer ist es ratsam, einen Gegner anzugreifen. An manchen kann man besser vorbei- oder drüberspringen. So ist der Ritter im 16. Screen unbesiegbar, der hinter im liegende Bogen nur eine Atrappe. Selbst am Drachen kann man mit dem richtigen Timing vorbeispringen. In späteren Levels kann man das Spiel auch sehr gut zu zweit spielen. Ein Spieler kämpft mit dem Joystick, der zweite bedient mit der Maus die Menüleiste. Selbstverständlich kann das Spiel auch komplett per Maus oder Tastatur gespielt werden. In der Endphase läuft die Zeit rückwärts, die Uhr zeigt dann an, wann der Vulkan ausbricht und die Spitze des Berges explodiert.

Sehr gut ist, daß der Programmierer daran gedacht hat, daß auch der hartnäckigste Barbar von einer ihm wohlgesonnenen Barbarin einmal abgelenkt werden kann - oder eine kleine PAUSE braucht.

Wenn Sie sich jetzt von Durgans Unterwelt genauso angezogen fühlen wie Hegor, schreiben Sie uns doch einfach eine Postkarte. Wir verlosen unter den Einsendern fünf Originale von "Barbarian" für den Atari ST.

Schicken Sie uns eine Postkarte bis zum 26. September 1987 (Datum des Poststempels). Es gelten die üblichen Wettbewerbsbedingungen, der Rechtsweg ist ausgeschlossen. Sollten Sie bei der Auslosung kein Glück haben oder nicht so lange warten wollen: Barbarian gibt es für ca. 69 Mark auch im Fachhandel oder direkt bei:

Psygnosis
Port of Liverpool Building
Pier Head
Liverpool, England
Tel.: 0044-51-236-7757

(cpl)





DIZZY WIZARD - wer schafft die Labyrinthe?



DIZZY WIZARD

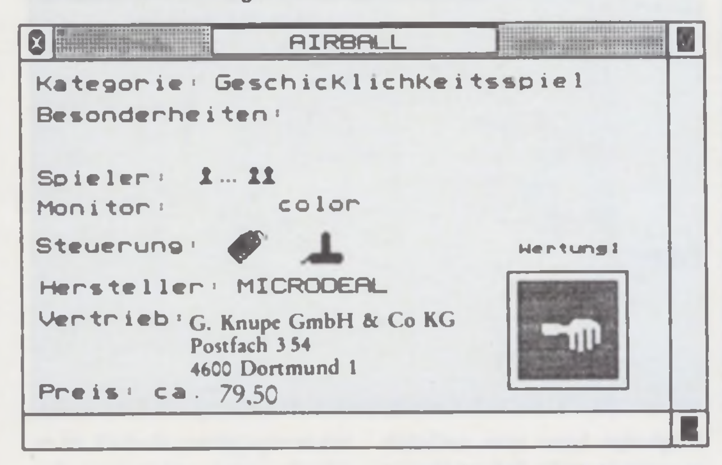
Kategorie: Geschicklichkeitsspiel
Besonderheiten: interessante Spielidee
Spieler: 1...3
Monitor: sw & color
Steuerung:
Wertung:
Hersteller: Tommysoftware
Vertrieb: Tommysoftware
Gutzkowstrasse 35
6000 Frankfurt 70
Preis: ca. 98.-



Ausgestattet mit einem Konto von 10.000 Punkten, müssen Sie eine Flamme suchen und dabei einen labyrinthartigen Weg durchlaufen. Dessen Stege sind manchmal sehr eng, aber ein Absturz kostet Sie nur einige Strafpunkte und etwas Zeit. Ihre 'Leben' sind praktisch unbegrenzt, das einzige Limit ist die Zeit, und die kann gerade einem Neuling recht schnell ausgehen. Denn richtig schwierig wird es erst, wenn man diese Flamme aufgenommen hat. Nun muß man den Weg wieder rückwärts gehen und die Flamme an einer bestimmten Stelle absetzen. Dann geht es in den nächsten Level und das Spielchen wiederholt sich, wobei sich allerdings das Labyrinth verändert hat. Auf dem Weg begegnet man manchen Gegenständen, mit denen ein Kontakt nicht immer vorteilhaft ist. So sollte man z. B. den springenden Bällchen aus dem Weg gehen, denn sie saugen nur am Punktekonto. Dagegen können Fragezeichen oder Würfel durchaus eine Hilfe sein, denn entweder gibt es Bonuspunkte oder es wird eine Übersicht des Labyrinths angezeigt. Wenn das Punktelimit nach Ablauf der Zeit unterschritten ist, dann müssen die fehlenden Punkte 'abgearbeitet' werden, der Spieler muß in die 'Maschine'. Eine bisher einmalige Option ist, daß drei Spieler gleichzeitig am Geschehen teilnehmen können. Gesteuert wird dann mit Maus, Joystick und Tastatur. Vor allem das Zusammenspiel der drei Mitspieler ist nun gefragt. Die 99 Levels von DIZZY WIZARD und das nicht gerade einfache Spielgeschehen bieten lange Spaß. Besonders lobenswert ist außerdem, daß das Spiel sowohl auf dem Color- als auch auf dem Monochrom-Monitor läuft.

(mn)

AIRBALL - Wem geht hier die Luft aus?



AIRBALL

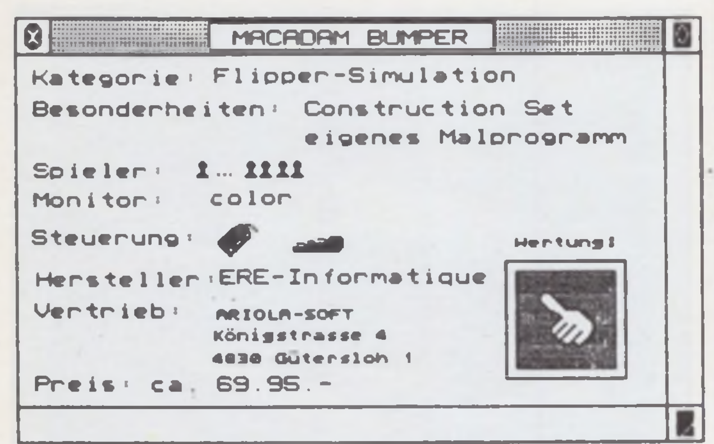
Kategorie: Geschicklichkeitsspiel
Besonderheiten:
Spieler: 1...2
Monitor: color
Steuerung:
Wertung:
Hersteller: MICRODEAL
Vertrieb: G. Knupe GmbH & Co KG
Postfach 3 54
4600 Dortmund 1
Preis: ca. 79,50



Ein hüpfendes Bällchen, oder eher ein Ballon, denn schon die kleinste Berührung mit einem spitzen Gegenstand zerstört die Außenhaut und das Ding saust davon. Natürlich hat auch diese Spiel eine Hintergrundstory: Ein böser Zauberer hat Sie in einen Ball verwandelt und in einem Schloß mit 150 Räumen ausgesetzt. Nun gilt es, auf der rastlosen Jagd durch die verschiedenen Levels, die verstreuten Gegenstände aufzunehmen und die Hindernisse zu bewältigen. Dabei geht Ihnen, im wahrsten Sinne des Wortes, die Luft aus. Deshalb muß immer wieder ein Zimmer mit einer Pumpe aufgesucht werden. Aber nicht übertreiben, denn zu viel Luft bringt Sie zum Platzen!
Es versteht sich von selbst, daß das Herumirren auch einen Sinn hat, denn es gilt, das Zauberbuch zu finden, dessen magische Sprüche Sie wieder in einen Menschen zurückverwandeln werden. Aber bis da hin ist es ein langer Weg. Jeder Raum hat seine Eigenheiten und kann nur mit einer bestimmten Taktik gemeistert werden.
Der Wechsel zwischen den Räumen geht sehr schnell und die Szenen sind recht abwechslungsreich. Die Steuerung ist eigentlich recht einfach, allerdings ist der Ball so empfindlich, daß man bisweilen sehr genau steuern muß. Ansonsten ist AIRBALL ein unterhaltsames und schnelles Spiel, bei dem man vor allem einen genauen Lageplan der Zimmer erstellen muß, um einen Überblick über die Örtlichkeiten und die Objekte zu haben.

(mn)

Für Flipper-Fans: MACADAM BUMPER



MACADAM BUMPER

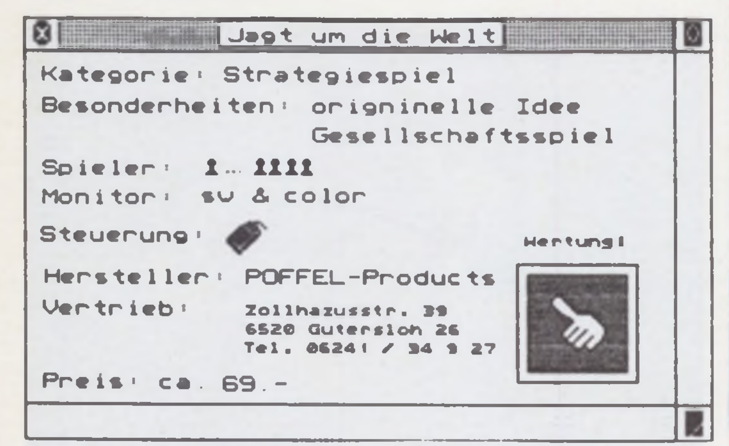
Kategorie: Flipper-Simulation
Besonderheiten: Construction Set
eigenes Malprogramm
Spieler: 1 ... 1111
Monitor: color
Steuerung:
Wertung:
Hersteller: ERE-Informatique
Vertrieb: ARIOLA-SOFT
Königsstrasse 4
4830 Gütersloh 1
Preis: ca. 69.95.-

Flipper haben schon immer eine besondere Faszination auf die Spieler ausgeübt. Eine französische Firma hat sich nun daran gemacht, dieses Feeling auf den Computer zu übertragen. Sicher wird ein passionierter Flipperfan sich nie von seinem Spielhallengerät trennen, aber trotzdem verspricht diese Simulation sehr viel Spielspaß. Der MACADAM BUMPER ist nicht nur ein Flippergerät, das man einfach spielt, sondern er ist ein 'Construction Set'. Jeder kann also seinen eigenen Flipper konstruieren, indem er Bauteile wie Bumper, Drop-Targets, Spinner, Discs und Flipper nach eigenen Wünschen anordnet. Ein kleines Malprogramm gestattet dann die weitere optische Gestaltung mit Linien und Farben. Leider hat man jedoch vergessen, daß viele auch das Titelbild des Flippers verändern wollen.

Nachdem der Aufbau des Flippers abgeschlossen ist, werden die typischen Einstellungen vorgenommen. Diese Betreffen einmal die Punkte, die beim Berühren der Objekte erzielt werden, außerdem die Bonuskombinationen. Besonders wichtig sind diejenigen Einstellungen, die Neigung, Geschwindigkeit und Elastizität der Kugel bestimmen. Um die richtige Einstellung zu finden, die dem originalen Flipper an nähesten kommt, benötigt man schon etwas Geduld, aber dann wird kräftig geflippert. Wer meint, daß die Konstruktion eines Flippers zu schwierig ist, der kann auch einen der mitgelieferten spielen, die zum Teil sehr gut aufgebaut sind. Eine ausführliche Demo zeigt außerdem die wichtigsten Schritte zum Aufbau. Der MACADAM BUMPER ist ein sehr gut aufgebautes 'Construction Set', mit dem man interessante Flipper bauen kann. Die Einstellmöglichkeiten sind sehr vielfältig und lassen kaum Wünsche offen. Also dann - viel Spaß beim Flippern!

(mn)

Jagd um die Welt



Jagt um die Welt

Kategorie: Strategiespiel
Besonderheiten: origninelle Idee
Gesellschaftsspiel
Spieler: 1 ... 1111
Monitor: sw & color
Steuerung:
Wertung:
Hersteller: POFFEL-Products
Vertrieb: Zollhazusstr. 39
6520 Gutersloh 26
Tel. 06241 / 34 1 27
Preis: ca. 69.-



Professor Uschnuk ist von den Schurken De Bleuchamps entführt worden. Der Professor hatte erst vor einigen Wochen seine sensationellen Forschungsergebnisse veröffentlicht, die, wenn sie in falsche Hände geraten würden, großes Unheil auf der Welt anrichten können.

Nun kommen Sie ins Spiel: Ihre Aufgabe ist es, De Bleuchamps zu erwischen und den Professor zu befreien. Es können 1 bis 5 Spieler mitmachen. Bei 3 bis 5 Spielern übernimmt jeder Mitspieler einen Agenten, bei 2 Spielern jeder 2 Agenten. Spielen Sie alleine, können Sie die Anzahl der Agenten selbst bestimmen. Die Agenten spielen nicht gegeneinander, sondern nehmen gemeinsam die Verfolgung auf. De Bleuchamps kann sich überall auf der Welt befinden. Die Agenten müssen versuchen, ihn so einzukreisen, daß er keine Fluchtmöglichkeit mehr hat.

Auf dem Bildschirm befindet sich eine Weltkarte mit unterschiedlich großen Flughäfen und Fluglinien. Von einem großen Flughafen kann man natürlich in weitaus mehr Städte fliegen als von einem kleinen, allerdings ist der Preis auch höher. So kostet zum Beispiel ein Flug von einem kleinen Flughafen zum nächsten, unabhängig von der Größe, immer 100 Dollar. Ein Flug zwischen zwei großen Flughäfen kostet schon 800 Dollar. Jeder Agent hat 7000 Dollar, und sollte er kein Geld mehr besitzen, so ist für ihn die Jagd gelaufen. Nach jeweils zwei Runden wird der Ort, an dem sich De Bleuchamps gerade aufhält, bekanntgegeben; deshalb ist es ratsam, seine Agenten zu diesem Zeitpunkt in der Nähe von Flughäfen zu postieren, von denen aus weite Strecken zurückgelegt werden können.

Jagd um die Welt ist ein spannendes Verfolgungsspiel, bei dem die ganze Familie mitjagen kann. Aber auch alleine kann man einfach nicht früher aufhören als De Bleuchamps geschnappt und der arme Professor befreit ist.

(AS)



Die CES ist nichts weniger als die größte Elektronik-Fachmesse der Welt: Treffender als mit diesem Superlativ kann man sie nicht beschreiben. Zweimal im Jahr tummelt sich ein ausgesuchtes Publikum (nur Elektronikfachhändler und Journalisten sind zugelassen) auf einer Ausstellungsfläche von mehr als 30 000 Quadratmetern bei mehr als 1400 vertretenen Firmen. Der Ausstellungsort wechselt im Halbjahresrhythmus: Die Sommer-CES findet stets in Chicago, die Winter-CES in Las Vegas statt.

Die großen Messen in Deutschland (CEBIT), England (PCW) und USA (CES) unterscheiden sich in ihrer Struktur wesentlich: Die PCW ist eine öffentliche Verkaufmesse. Auch die CEBIT ist öffentlich, doch keines der dort ausgestellten Produkte kann sofort gekauft und mitgenommen werden. Die CES geht noch einen Schritt weiter: Wie erwähnt, ist sie nicht für jedermann zugänglich, und auch direkt mitnehmen kann man nichts. Es bleibt also nur die Bestellung, und auch die nur bei größeren Stückzahlen. Dennoch zählt die CES regelmäßig über 100 000 Besucher. Dies liegt daran, daß sie nicht auf Computer beschränkt ist. So gab es in der Zeit zwischen 30. Mai und 2. Juni etwa vierzig Workshops zu unterschiedlichen Themen, von „New Dimensions in Retail Sales Training" bis hin zu „The Management of Family-Owned Business". Im Bereich Photographie wurde eine Kamera vorgestellt, die Bilder auf Diskette abspeichert. Sie schafft fünf Bilder pro Sekunde, wiegt rund ein Kilo und soll etwa tausend US-Dollar kosten. Erhältlich für Händler – ab Herbst. Für Musikfreaks gibt es jetzt den Plattenspieler mit visco-elastischem Material, das unerwünschte mechanische Vibrationen auffängt, Videofans können ab Herbst auf das CD-Video zurückgreifen, und wer von einem reinen Spielcomputer in Spielhallenqualität träumt, findet die entsprechende Konsole bei Nintendo.

Auch für die Atari-ST-Besitzer wurde natürlich einiges geboten. Der Atari-Stand selbst war unterteilt in zahlreiche kleinere Stände von Unterausstellern, z. B. Microdeal und Timeworks.

An erster Stelle der in Chicago für den Atari ST präsentierten Neuheiten standen Spiele. Das liegt sicher daran, daß der Atari ST in den USA weit weniger als Anwendungscomputer betrieben wird als hierzulande. Gute Anwender-Software war auf dieser Messe kaum zu finden. Ebenso spärlich waren neue Drucker oder andere Hardware zu finden. Einzig Epson präsentierte mit dem LX 800 einen 9-Nadel-Drucker in guter Qualität und zum günstigen Endpreis von 269 US-Dollar, der ab August erhältlich sein soll. Vorab die technischen Daten:

Schrift	Elite	180	Zeichen pro Sec.
Schrift	Pica	150	ZpS
NLQ	Elite	30	ZpS
NLQ	Pica	25	ZpS

Character Set:
96 ASCII Characters
96 Italic Characters
32 international Characters
32 Italic int. Characters
32 Epson Graphics Characters
96 IBM Graphics Characters

Der LX-800 verfügt über Einzel- und Endlospapiereinzug, hat 3 K Buffer und kann über Centronics betrieben werden. Wir werden das Gerät in einer unserer nächsten Ausgaben testen.

Software, meist Spiele, gab es reichlich. Die folgende Aufzählung der interessantesten Neuvorstellungen erfolgt deshalb, der besseren Übersicht wegen, alphabetisch:

ACCOLADE

Hardball:
Die Umsetzung des bereits vom C 64 bekannten Baseballspiels.

Testdrive:



Mit dem Ferrari oder Lamborghini geht die Testfahrt quer durch die USA – vorbei an Trucks, herabgestürzten Felsen und Baustellen und immer auf der Hut vor den Polizeistreifen.

EPYX

CSub Battle Simulator:



Ein U-Boot-Abenteuer im II. Weltkrieg.

Spy vs. Spy III, Arctic Antics:
Spion und Spion kloppen, bomben und sprengen sich diesmal durch das ewige Eis auf der Suche nach Raumhelm, Navigationskarte und anderen Utensilien, um das Raumschiff flott zu machen und der Eiswüste zu entrinnen. Leider ist es wieder mal ein Einmannschiff und nur der hinterhältigere wird entkommen.

Boulder Dash Construction Kit:
Das kleine Männchen, das so heldenhaft Diamanten sammelt, kennt jeder bereits vom C64. Mitsamt einem dazugehörigen Construction-Kit ist es jetzt auch in den Speicher des Atari eingezogen.

FIREBIRD

The Advanced OCP Art Studio:
Ein Zeichenprogramm mit 16 verschiedenen, vom Benutzer definierbaren Brushes, Zoom u. a. m.

Universal Military Simulator:
Ein Kriegssimulator, der keine Wünsche offenläßt. Der Spieler kann sich sein Schlachtfeld und alles, was dazugehört, selbst erstellen. Ob dieses Programm allerdings die Schlacht mit der Zensur übersteht, bleibt abzuwarten.

Knight Orc:
Eine Graphik-Adventure-Trilogie. Ein Abenteuer, bei dem Sie sich als Orc unter die Menschen mischen – das richtige Adventure für Freunde des makabren Humors.

HYBRID ARTS

Midi Maze:
Das Spiel, das auf bis zu 16 Atari ST-Rechnern gleichzeitig gespielt werden kann, alle verbunden durch ein MIDI-Kabel.

JAGUARE

Alien Fires:
Ein Acton-Strategie-Spiel zweier junger kandischer Programmierer. Im nächsten Heft stellen wir die deutsche Version davon vor. Jeff Simpson hat uns zehn Exemplare für die ST Computer-Leser zur Verfügung gestellt.

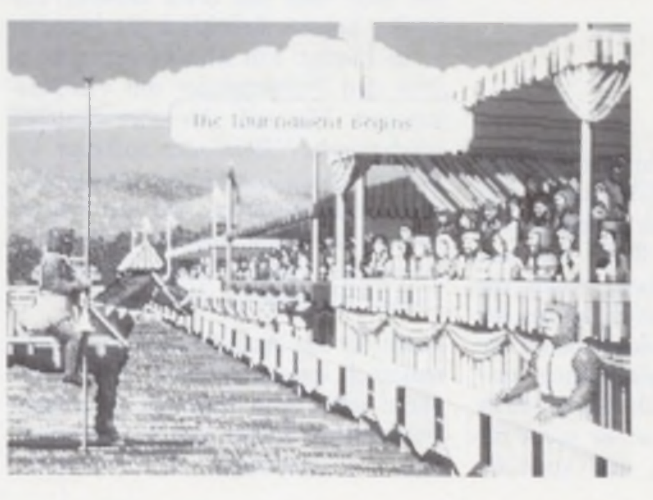
MASTER DESIGNER

Defender of the Crown:
Rittergetümmel aus der Zeit von Robin Hood und den Angelsachsen – ab Herbst kommt Cinemaware damit auch in Ihr Heimcomputerkino.



Defender of the crown:

Wie das Leben so spielt, sind auch Intrigen an der Tagesordnung.



Turniere sind das Salz in der Suppe eines richtigen Ritters, der immer auf der Suche nach Ruhm und Ehre ist.

King of Chicago:





Das Neueste aus der Hitschmiede des Cinemaware-Teams. Ein Spiel um die Nachfolge des Gangsterkönigs Al Capone in Chicago mit animierter Graphik. Erhältlich ab Ende August für den Amiga; die ST-Version soll bald darauf folgen.

MINDSCAPE

Into the Eagles Nest:
Ein vom C64 bekanntes Spiel, das im zweiten Weltkrieg spielt. Der Held muß aus einer Basis der Nazis drei Gefangene befreien. Die Umsetzung für den Atari ST wird aber erst Ende des Jahres im Handel sein.

PASSPORT DESIGN

Midisoft Studio:
Ein weiteres Midi-Programm für den ST. Es bietet Platz für mehr als 80.000 Noten pro Song und hat 32 unabhängig verwaltete polyphonische Spuren. Der Preis liegt bei 99 US-Dollar.

PSYGNOSIS

Barbarian:
Ein Höhlenspiel aus grauer Vorzeit. Alles weitere im Spieleteil.

STAR SOFT INTERNATIONAL

The Pirates of the Barbary Coast:
Dies ist der Titel des ersten Spiels von SSI. Die vom Atari XL bekannte englische Softwarefirma „RED RAT" ist bei SSI beteiligt. Das Spiel selbst werden wir in einem der nächsten Hefte besprechen. Es soll im Herbst erscheinen.

TIMEWORKS

Word Writer ST:
Ein Textverarbeitungsprogramm in englischer Sprache mit einem englischen Lexikon und Synonymen für über 60.000 englische Wörter.

Partner ST:



Ein Büroprogramm mit Textverarbeitung, Kalender, Notizblock, Telephonspeicher u. a. m. Alles in englischer Sprache und für den amerikanischen „Freizeitmanager" gedacht.

Am Rande der Messe, sozusagen inoffiziell, gab es dann noch die eine oder andere Information von Firmen oder Programmierern zu erhaschen. So plant MicroProse eine Umsetzung verschiedener Simulatoren; Avalon Hill, Hersteller bekannter Strategiespiele wie 'Diplomacy', 'Spitfire 40' und 'Dnieper River Line', wird demnächst seine Produkte für den Atari ST umsetzen. Atari selbst gab eine mehrer Zentimeter dicke Pressemappe heraus. In ihr war jedoch – den ST betreffend – gegenüber der CEBIT 87 nichts Neues zu finden. Der Rest der Mappe war der „neuen agressiven Marktstrategie", den – wie bei jeder Firma – dazugewonnenen Marktanteilen und der Vorstellung einiger erwählter Häupter gewidmet. Dem von der größten Computermesse der Welt verwöhnten CEBIT-Besucher konnte die CES diesmal nicht als „der ganz große Wurf" erscheinen. Was bleibt, ist die Vorfreude auf die PCW (Ende September in London), wo die meisten der hier angekündigten Programme dann auch erhältlich sein sollen.

(cpl)

DAS BLITTER-TOS GEHT UM
Patch für GfABASIC

Kaum hat man sich an die terroristischen Aktivitäten der alten ROMs (Bombenanschläge etc.) gewöhnt, taucht schon eine neue Version des TOS auf. Es wird vor allem in den MegaSTs gebraucht, die damit den Blitter ansteuern (wann endlich gibt es den Blitter für die alten STs??? Seufz...). Aber wir haben Glück: Wir müssen uns nicht an ein langweiliges, fehlerloses neues TOS gewöhnen; vielmehr gewinnt auch diese Version einen ganz typischen Charakter durch ihre erfrischenden Eigenwilligkeiten. Dazu kommt, daß viele Progamme nicht mehr laufen oder zumindest nicht so wie gewohnt. Unter diesen auch GfABASIC...

GfABASIC und das Blitter-TOS leben zwar recht friedlich nebeneinander her, doch wenn man versucht, ein GfABASIC-Programm – wie man es gewohnt ist – mit der Tastenkombination ALTERNATE-CONTROL-SHIFT zu stoppen, zeitigt das ganz unerhörte Wirkung, nämlich gar keine. Systemabsturz? Kaputte Tastatur? Mitnichten. GfA-Programmierer Ostrowski hat nur etwas unsauber auf undokumentierte Systemvariablen zugegriffen, deren Lage sich im neuen TOS verschoben hat.

Im „alten" ROM-TOS vom 6.2.86 merkte sich die Tastaturroutine den Zustand der Sondertasten (wozu auch ALTERNATE, CONTROL und SHIFT gehören) in der Speicherstelle $E1B. Das nutzte das GfABASIC weidlich aus, anstatt sich an die vorgeschriebenen Wege über die KBSHIFT-Funktion des BIOS zu halten. Im neuen TOS vom 22.4.87 steht diese Variable aber bei $E61 – und schon läuft die Stopabfrage nicht mehr.

Nach einer intensiven Suchaktion habe ich die Stelle gefunden, die im GfABASIC abzuändern ist. Das Programm BLTPATCH.BAS (Listing 1) erledigt die Modifikation automatisch. Die gepatchte Version ist allerdings auch nur auf dem Blitter-TOS (neue Version) voll lauffähig – auf der alten ROM-Version funktioniert die Stopabfrage nicht mehr. Deswegen sollten Sie sich eine Kopie des Originals aufheben und die gepatchte Version umbenennen. Übrigens funktioniert der Patch nur bei der Version 2.0 von GfABASIC.

Ein Punkt läuft Amok

Damit ist es aber noch nicht getan. GfABASIC hat die hilfreiche Angewohnheit, beim Speichern von MERGE-Files (Extension .LST) und Programmdateien (Extension .BAS) die Extension selbständig anzufügen, wenn man sie nicht eintippt. Beim Experimentieren unter Blitter-TOS fiel auf, daß das nicht mehr funktioniert, wenn der Dateiname genau 8 Zeichen lang ist. Zuerst vermutete ich einen Fehler im GfABASIC, doch wie sich herausstellte, ist Ostrowski unschuldig und braucht dieserhalb und desterwegen nicht gelyncht zu werden.: Im neuen ROM ist (mindestens) ein Fehler!

Die Fileselect-Box (das ist der im GEM vordefinierte Dialog zum Bestimmen des Dateinamens beim Einladen und Abspeichern) hat bei den Aufräumungsarbeiten eine Macke abbekommen: Tippt man weniger als acht Zeichen ein, ist alles in Ordnung (Beispiel: Man gibt 'TESTDAT' ein, und die Fileselect-Funktion liefert dann auch den String 'TESTDAT'). Ist der eingegebene Dateiname aber gerade acht Zeichen lang, zum Beispiel 'TESTFILE', bekommt man 'TESTFILE.' von der Fileselect-Routine zurück (man beachte den Punkt nach dem Dateinamen)! Das ist nun aber so falsch, wie es nur sein kann. Im GfABASIC steht nun offensichtlich eine Abfrage auf die Länge des Rückgabestrings der Fileselect-Box. Ist dieser String weniger als neun Zeichen (also höchstens acht Zeichen) lang, ergänzt das Programm die Extension (.LST oder .BAS je nach Art des Abspeicherns); ist der String länger, nimmt GfABASIC an, daß der Anwender selbst eine Extension eingetippt hat und beläßt es dabei. Durch den fehlerhaft zurückgelieferten Punkt im neuen ROM greift diese Abfrage schon, wenn man nur acht Zeichen eingegebn hat – weil der Punkt zusätzlich mitgeliefert wird.

Abhilfe: Extension mit eintippen. Oder darauf warten, bis entweder Ostrowski seinem BASIC eine Krücke baut (was nicht viel helfen würde, denn vom ROM-Fehler sind ja auch andere Programme betroffen), oder bis ich mich im ROM-Code zur fehlerhaften Stelle durchgewühlt habe.

Wenn Sie selbst Fehler im neuen TOS gefunden oder gar schon gepatcht haben, schreiben Sie doch mal der ST-Computer.

Claus Brod

```
Patch für GfABASIC                Seite 1          (C) 1987 Claus Brod

' Patch für GfABASIC 2.0
' für Blitter-TOS
' Beseitigt Fehler bei ALT-CTRL-SHIFT-Abfrage
' Found and written 9.8.1987 by
'               Claus Brod
'               Am Felsenkeller 2
'               8772 Marktheidenfeld
'               09391/3206
'
Alert 1,"Patch für GfABASIC|auf Blitter-TOS",1,"OK|Raus!",A
If A=1
  Open "U",#1,"\gfabasic.prg"
  Seek #1,&H63AD
  Out #1,&H61
  Close #1
  Print "Fertig - GfABASIC für Blitter-TOS gepatcht"
Endif
```



Geschwindigkeit ist Trumpf:

Der OMIKRON-Basic Compiler sticht

Der bekannte OMIKRON-Basic Interpreter, der allein schon der schnellste Basic-Interpreter für den ST ist, wurde nun mit dem Compiler bereichert. Dies macht es möglich, daß man mit Basic in Geschwindigkeitsbereiche vordringen kann, die bisher nur der Sprache C vorbehalten waren.

Ein Compiler muß her

Bei vielen Sprachen handelt es sich ausschließlich um Compilersprachen, z. B. Pascal, C, Fortran oder Modula 2. Das hat zwar den Vorteil, daß das fertige Programm sehr schnell im Ablauf wird, doch dauert es natürlich einige Zeit, bis ein Programm fertig übersetzt ist. Ideal ist es natürlich, wenn man ein Programm auf einem Interpreter entwickelt, da es dort sofort läuft, man also die bei großen Programmen recht langen Übersetzungszeiten sparen kann, und es dann erst compiliert. Genau dies ist mit dem OMIKRON-Basic-Entwicklungspaket möglich. Dies ist momentan die eleganteste Möglichkeit, Programme zu entwickeln.

Die Zeit läuft: Compiler ★ 10 = Interpreter

Was schon beim Interpreter eine recht große Rolle spielte, nämlich die Ausführungszeiten eines Programms, nimmt bei einem Compiler eine noch größere Bedeutung ein. Die Leute von OMIKRON durften sich bei ihrem Interpreter bekanntlich nicht beklagen, denn er ist und bleibt der schnellste Interpreter für den Atari ST. Dennoch stellt der Compiler sein Mutterprogramm fast in den Schatten. Der erreichte Geschwindigkeitszuwachs beträgt im Durchschnitt rund den Faktor 10. Bei bestimmten Befehlen, besonders bei der Integerarithmetik, sind Faktoren von 20 oder mehr keine Seltenheit.

...und der Rest der Welt

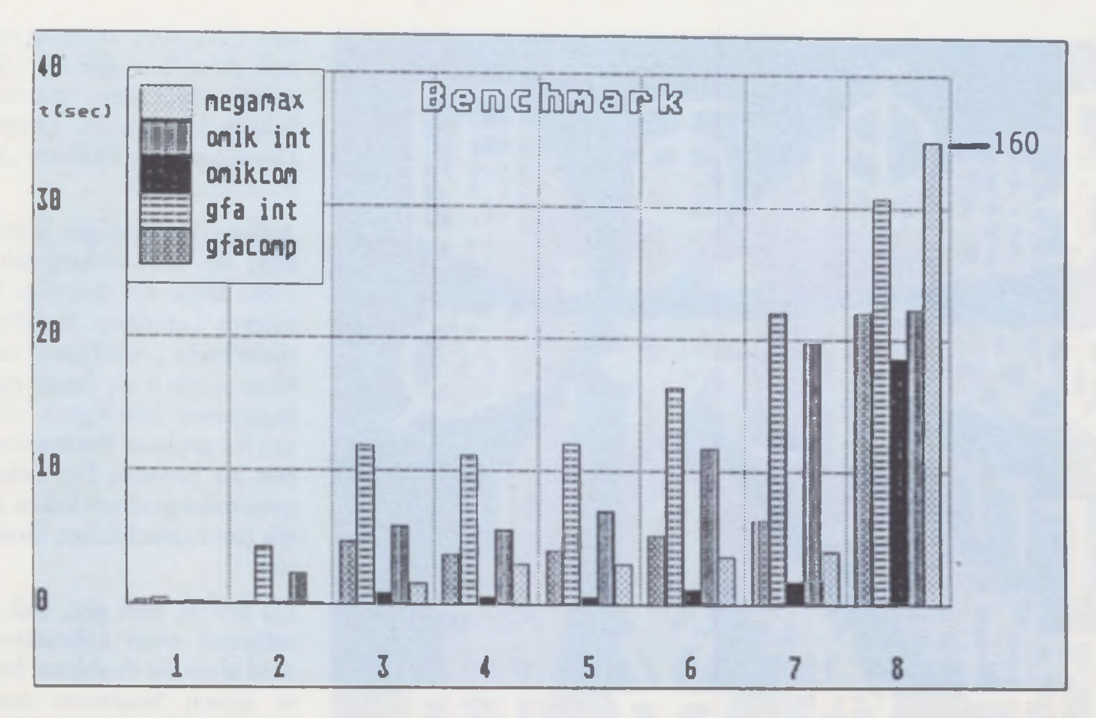
Da es auch noch andere Compiler für den ST gibt, ist natürlich der direkte Vergleich mit ihnen von Interesse. Auf Bild 1 sehen Sie die gemessenen Zeiten von OMIKRON- und GFA-Basic im Vergleich mit dem Megamax C-Compiler. Das Ergebnis dürfte recht verblüffend sein, da der OMIKRON-Compiler sogar bessere Zeiten als der C-Compiler erreichte. Dazu muß man allerdings sagen, daß bei Megamax ohne Registervariablen gearbeitet wurde, da diese den Basic-Compilern noch unbekannt sind; OMIKRON-Basic steht diese Erweiterung demnächst ins Haus.

Da Benchmarktests immer etwas kritisch sind in Bezug auf das gleiche Zahlenformat, haben wir uns auf Long-Integer geeinigt, da z. B. bei Float schon allein das unterschiedliche Format keine genauen Schlüsse zuläßt. Daß Benchmarks in dieser Form, gerade zwischen verschiedenen Programmiersprachen, ohnehin nicht unumstritten sind, möchten wir hier nicht verschweigen. Im Vergleich mit dem direkten Basic-Konkurrenten (GFA-Basic) sieht man jedoch, daß der OMIKRON-Basic-Compiler, wie schon der Interpreter, der schnellere ist. Das ist umso verwunderlicher, als der OMIKRON-Basic-Compiler, im Gegensatz zum GFA-Basic-Compiler, nicht in Assembler, sondern selbst in OMIKRON-Basic geschrieben wurde und sich schließlich selbst übersetzte.

Compiler < = > Interpreter?

Ein sehr wichtiger Punkt ist natürlich die Kompatibilität des Compilers zum Interpreter. Hier gibt es einige kleine Unterschiede, die sich allerdings auf das Dimensionieren von Feldern und Beachten des Zahlenformats bei Funktionsaufrufen beschränkt. Dies ist aber aufgrund eines optimalen Codes nicht anders machbar; ein Pascal-Compiler ist da wesentlich pingeliger. In solchen Fällen gibt der Compiler jedoch eine genaue Meldung aus, mit der man Fehler leicht ausbügeln kann.

Als Besonderheit des Compilers tritt in Erscheinung, daß er innerhalb des compilierten Programmes dynamische Felder verwalten kann, also im Programmablauf ein Feld redimensionieren (vergrößern und auch verkleinern)



kann; eine praktische Eigenschaft, beispielsweise für Dateiverwaltungen.

Zusätzlich weist er noch Optionen auf, mit denen man in Regionen vordringen kann, die einem Basicprogrammierer bisher fremd waren.

Das Zauberwort 'Multitasking'

Der im Interpreter bereits bestehende 'ON TIMER GOSUB'-Befehl wurde verfeinert: Er springt im Interpreter nach Ablauf einer bestimmten Zeit und nach Vollendung eines Befehls in eine Prozedur, führt diese aus und kehrt von dort in das Hauptprogramm zurück. Da dieser Sprung aber nur nach jedem vollendeten Basic-Befehl möglich wurde, konnte die Interruptroutine nicht gleichmäßig angesprungen werden. Bei INPUT beispielsweise war der Interrupt blockiert und bei zeitaufwendigen Befehlen (z. B. SORT, MAT INV) erfolgte er mit viel Verspätung. Im Compiler ist es nun möglich, auch mitten in einem solchen Befehl in das Unterprogramm zu verzweigen. Damit lassen sich Echtzeit-Anwendungen erstellen, die sich durch nichts aufhalten lassen.

Die Steuerworte, die die Optionen festlegen, können während des Programmablaufs verändert werden. Dadurch kann man Unterbrechungen (z. B. Ctrl-C von der Tastatur oder eigene Interruptroutinen) auf bestimmte Programmteile beschränken.

Anleitung

Die Anleitung enthält die möglichen Fehlermeldungen und sagt, wie man sie behebt. Ferner werden dort einige Tricks beschrieben, um Programme noch zu beschleunigen. Ein weiterer Pluspunkt der Anleitung ist, daß sie einige interne Informationen gibt, beispielsweise die Speicherverwaltung des Compilers, was besonders fortgeschrittene Programmierer interessieren dürfte.

Der Interpreter

Die Leistungsfähigkeiten eines Compilers sind natürlich auch stark von denen des Interpreters abhängig, deshalb wollen wir ihn hier kurz beschreiben.

OMIKRON-Basic arbeitet mit oder ohne Zeilennummern, ganz nach eigenem Geschmack, erlaubt pascalähnliche, strukturierte Programmierung durch Prozedur- und mehrzeilige Funktionsaufrufe mit Parameterhin- und rückgabe. Trotz der vielen, den Basic-Standard ergänzenden Befehle ist OMIKRON-Basic noch kompatibel zu M-Basic, d. h. man kann diese Programme einfach überspielen und kann, was vor allem für Anfänger wichtig ist, mit gängigen Lehrbüchern arbeiten.

In Sachen Genauigkeit ist OMIKRON-Basic mit seinen doppelt genauen Zahlenformaten bei allen Operationen (19 Stellen) bisher unübertroffen. Im mathematischen Bereich glänzt es durch seine Vielzahl von Befehlen, die einerseits den kompletten trigonometrischen und hyperbolischen Bereich abdecken und andererseits dem Anwender Befehle für Matritzen-Berechnung zur Verfügung stellen, mit denen die Programmablaufgeschwindigkeit wesentlich vergrößert wird.

Alle GEM-Befehle sind in einer Bibliothek eingebaut, d. h. man kann genauso programmieren wie in C.

Fazit

Der Compiler hält das, was man von ihm erhoffte. Nach kurzer Einarbeitungszeit zeigten sich keine Probleme mit dem Übersetzen von Basic-Sourcefiles in Assembler-Programme. Die Geschwindigkeit der übersetzten Programme stellt manche andere Compilersprache in den Schatten und läßt fast die Frage aufkommen, ob man Betriebssysteme demnächst, wenn schon in C statt Assembler, nicht auch in Basic schreiben könnte.

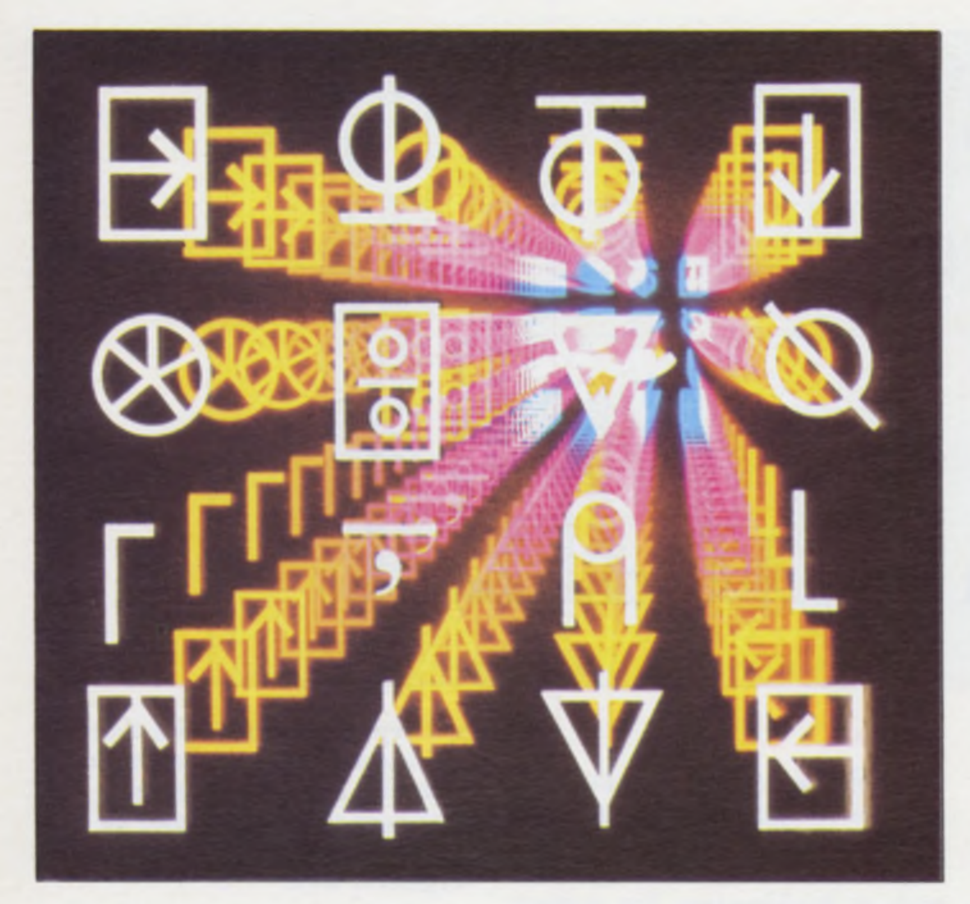
Die neuen, Multitasking betreffenden Optionen des Compilers bereichern das Paket um eine weitere interessante Anwendungsmöglichkeit.

Bleibt zu sagen, daß der OMIKRON Basic-Interpreter zusammen mit dem Compiler das beste Basic-Entwicklungssystem für den Atari ST darstellt.

(HS)

Preis: 159,– DM

OMIKRON-Software
Erlachstr. 15
D-7534 Birkenfeld 2
Tel. 0 70 82 / 53 86



Zeichensprache

Schon mal was von APL gehört? Vielleicht nicht, denn diese Programmiersprache gilt vielen als exotisch.

Nun hat sich diese Zeitschrift entschlossen, das bestgehütete Geheimnis der Computerbranche zu lüften: Tatsächlich gehört APL zu den Top 3 der Programmiersprachen (im kommerziellen Bereich) und erfreut sich ständig steigender Beliebtheit.

Das hat Gründe. Der wichtigste ist wohl, daß APL keine Programmiersprache im herkömmlichen Sinn ist, sondern (man verzeihe mir diesen Ausdruck) ein „Denkwerkzeug". Der Unterschied wird am ehesten bei einem kleinen Rückblick in die Computer-Steinzeit deutlich:

Am Anfang wurde mit Schaltern und Stöpseln programmiert. Mit dem Erscheinen der Lochkarte besserten sich die Arbeitsbedingungen der Programmierer nur unwesentlich; sie mußten weiter in Nullen und Einsen denken, denn der Rechner interessierte sich nur für zwei Dinge: „Loch" oder „Nichtloch". Da war es schon ein entscheidender Fortschritt, als Befehle endlich in Form von englischen Sprachfetzen geschrieben werden durften. Die Listings waren zwar noch immer von beachtlichem Gewicht, aber man konnte sie leichter „lesen" als die Zahlenkolonnen der Frühzeit. Der nächste Schritt war, mehrere Rechnerkommandos zu einem Befehl zusammenzufassen und Kontrollstrukturen einzuführen. Und das war's dann auch schon.

Auch heutige Rechner arbeiten brav einen Befehl nach dem anderen ab und lösen jede Aufgabe in kleinste Schritte auf. Herkömmliche Programmiersprachen spiegeln diese Vorgehensweise wider. Um eine Problemstellung auf den Computer zu bringen, muß man sich detailliert mit der „Denkweise" seines Rechners auseinandersetzen: Schritt für Schritt, Loop für Loop. Das eigentliche Problem „versteht" die Maschine nicht.

Anfang der sechziger Jahre begann bei IBM die Entwicklung der legendären /360. Diese für damalige Verhältnisse extrem komplexe Maschine ließ sich nicht mehr „von Hand" entwerfen. Es hätte einfach zu lange gedauert. Die Ingenieure benötigten eine Sprache, um die geplante Rechnerlogik beschreiben zu können. Die bekannten Programmiersprachen haben aber mit Logik (im menschlichen Sinne) wenig zu tun.

Da traf es sich gut, daß ein Kollege während eines Lehrauftrags in Harvard ähnliche Probleme hatte: Er wollte seinen Studenten mathematische Logik beschreiben und hatte eine dafür geeignete Schreibweise entwickelt: APL. Es zeigte sich, daß mit dieser Notation auch Parallelprozesse des geplanten Rechners darstellbar waren.

Zu diesem Zeitpunkt war APL eine reine „Tafelsprache": Kein Computer kann mit Kreide programmiert werden. Aber die Entwickler bei IBM waren von ihrem neuen Werkzeug so angetan, daß sie es in der Folgezeit auf einen Rechner umsetzten. APL (bzw. seine Vorstufen) war in der Firma bald so begehrt, daß es schließlich 1968 auch für Kunden zugänglich gemacht wurde.

Die Entwicklungsgeschichte verdeutlicht, daß mit APL zum ersten Mal eine Notation menschlicher Denkweisen auf den Computer gebracht wurde, statt wie üblich die Computerlogik menschlicher zu verpacken. Was das im Einzelnen bedeutet, ist für Leute, die ständig in Schleifen denken müssen, allerdings nur schwer nachvollziehbar.

Datenstrukturen

Die Befehle eines APL heißen Funktionen. Sie sind allesamt dafür konzipiert, auf Strukturen zu operieren, das heißt, auf Datengebilden mit beliebig vielen Elementen. Das können sowohl Einzelelemente („Skalare") wie auch Vektoren oder Arrays („Matrizen") mit mehreren Dimensionen sein. Diese „Variablen" heißen nicht nur so, sie sind auch tatsächlich flexibel: APL

kennt keine Deklarationen oder feste Größen von Datenstrukturen.

Jede Variable kann jederzeit gebildet werden (durch Zuweisung), vergrößert oder verkleinert werden (z. B. durch Anhängen, Einfügen, Entfernen), eine andere Struktur bekommen oder den Datentyp wechseln. Apropos Datentyp: Für den Programmierer braucht es nur die Unterscheidung zwischen Zahlen und Text – den Rest erledigt APL. Intern arbeitet APL mit vier Typen: Text (im Byte-Format), Integer (meist vier Bytes pro Wert), Fließkomma (meist im IEEE-Format mit 64 Bit – „Double Precision") und Boolean (1 Bit pro Wert).

Es ist schon sehr angenehm, beim Programmieren nicht mit den Feinheiten der Rechner-internen Datenrepräsentation herumbasteln zu müssen: Bei manchen Berechnungen weiß man nicht von vorneherein, ob reelle Zahlen auftauchen werden. APL legt die Werte nach Bedarf ab; kommen nur ganze Zahlen vor, wird halt platzsparend im Integerformat abgelegt, gibt es plötzlich Nachkommastellen, wechselt die Maschine zum Float-Format. Daher muß man auch nicht festlegen, mit welcher Genauigkeit gerechnet werden soll: APL merkt von selbst, wenn es krumme Werte gibt.

Dies alles soll übrigens nicht heißen, daß Sie keine vollständige Kontrolle über die Details hätten. Man kann in APL die Rechengenauigkeit einschränken; man kann bestimmte Formate forcieren und in den Bits eines Characters wühlen – man muß es aber nicht, wenn man den Kopf für das Wesentliche freihalten will.

Programmieren mit „Icons"

Das Datenhandling ist also schon recht „menschenfreundlich" geregelt. Wie aber sieht es mit der Programmierung aus? Als ST-Computer-Leser sind Sie sicherlich an die GEM-Oberfläche des Atari-Betriebssystems gewöhnt. Wenn Sie aber nun ein Programm in einer der üblichen Sprachen schreiben wollen, hat es sich auch schon mit den Icons. Stattdessen müssen halbe Romane in einer Art „Pidgin-Englisch" in den Editor gehackt werden. Um die Übersicht zu behalten, ist alle naslang ein Ausdruck fällig – die meisten Subroutinen passen nicht auf eine Bildschirm-Seite.

APL braucht kein Englisch. Es hat Symbole. Sicher, die sind nicht so schön wie die GEM-Icons, aber immerhin haben die meisten von ihnen eine augenfällige Bedeutung. Jedes Symbol wird mit einer Taste aufgerufen und ist genau so groß wie ein Character: Das ist ein APL-Befehl. Keine langen Wörter, keine reservierten Ausdrücke, kein Pseudo-Englisch. Für eine in den sechziger Jahren entwickelte Sprache war das revolutionär. Aber im Zeitalter der grafischen Benutzeroberflächen und der Icons fragt man sich doch, ob die anderen Sprachen noch auf der Höhe der Zeit sind. Diese ketzerische Frage ist allerdings nicht so ganz ernst zu nehmen: Natürlich haben die verschiedenen Sprachen ihre Berechtigung. Nur wenn ich zum Beispiel sehe, daß komplexe Anwendungen, die nicht massenhaft verbreitet werden, weil sie zu speziell sind, in C programmiert werden, kommen mir die Tränen. So eine Mühsal ist doch unbezahlbar!

APL-Befehle sind aber nicht nur kompakt, sie lassen sich auch verknüpfen. Damit passen in eine Programmzeile komplexe Ausdrücke oder Berechnungen – so wird ein Programm kurz und übersichtlich. Um mehrere Kommandos aneinanderhängen zu können, braucht es nur eine einzige Regel: Rechts vor links. Das heißt, ein APL-Ausdruck wird immer von rechts nach links aufgelöst, es sei denn, man verwendet Klammern.

Der Effekt ist verblüffend: Die meisten mathematischen Ausdrücke können wie in der Schule notiert werden. Bei den Grundrechenarten („Punktrechnung vor Strichrechnung") stimmt das zwar nicht, aber begeben Sie sich mal in die höheren Gefilde der Algebra! Die oft verwirrenden (und meist willkürlichen) Regeln anderer Sprachen gibt es nicht. Und: Eigene Routinen folgen derselben Syntax wie die eingebauten Funktionen. Damit lassen sich beide in einem Ausdruck mischen, ohne daß es zu unerwünschten Seiteneffekten kommt.

Funktionen

Die Übereinstimmung von APL-Funktionen und eigenen Routinen geht bis in's Detail: Selbst die Bezeichnung ist identisch. Routinen (Module) werden ebenfalls „Funktionen" genannt. Ob Sie also schreiben

```
A+B
```

oder eine eigene Additionsfunktion „plus" benutzen

```
A plus B
```

ist beliebig. Im Beispiel sind „A" und „B" jeweils Variablen. Dabei ist es gleichgültig, ob es sich um Skalare, Vektoren oder Matrizen handelt: Der Ausdruck muß nur **„wohlgeformt"** sein, das heißt, jedem Element in A muß ein Element in B entsprechen. Ausnahme: Wenn eine der Variablen aus nur einem Wert besteht, wird dieser zu allen Werten der anderen Variablen addiert.

```
      1+4 7 9                    Ausdruck
5 8 10                           Ergebnis

      1 2 3+4 5 6
5 7 9
```

Aber APL kann nicht nur 1 und 1 zusammenzählen. Zu den üblichen Multiplikations- und Divisions-Funktionen kommen hinzu: Logarithmus, Reziprokwertberechnung, die Modulo-Funktion, Pi (mal X), Binominalkoeffizient, Fakultät, trigonometrische Funktionen... APL, das Mathe-Genie. Kein Wunder, wurde es doch von einem Mathematiker formuliert. Aber die Aufzählung ist unvollständig. Selbstverständlich gehören Vergleiche (größer, kleiner, größer-gleich, kleiner-gleich, ungleich, gleich), logische Funktionen (AND, OR, NAND, NOR und NOT), Aufrunden, Abrunden, Negieren, Potenzieren, und ein programmierbarer Zufallszahlengenerator zum mathematischen Sprachumfang (die Vergleiche gelten auch für Text).

Die Möglichkeit, ganze Matrizen am Stück zu „verwursten", läßt die Frage nach der Fähigkeit aufkommen, lineare Gleichungssysteme zu formulieren: Das geht. Die Matrixinversion ist als APL-Befehl vorhanden. **Ein** Zeichen. In Pascal braucht das entsprechende Programm drei Druckseiten, in denen vor allem mit den Unzulänglichkeiten des Computers herumgemacht werden muß. Der eigentliche Zweck des Pascal-Listings geht höchstens aus den Kommentaren hervor. Das APL-Symbol dokumentiert sich hingegen selbst: Es ist ein Rahmen (für „Matrix") mit einem Divisionszeichen darin. Kein Schrägstrich! Ein **Divisionszeichen**, wie in der Schule. Bei der Multiplikation sieht man auch keine Sterne, sondern ein kleines Kreuzchen. Warum sollen **wir** wegen eines dummen Computers

APL und "KI"

APL ist nicht unter der Vorgabe entstanden, "künstliche" Intelligenz zu produzieren. Angesichts dessen, daß schon die menschliche Intelligenz weitestgehend unerforscht ist, halte ich den Anspruch auf eine Nachbildung im Elektronengehirn für Quatsch. Aber: Die Methoden, die unter der (falschen) Flagge "KI" propagiert werden, gibt es unter APL auch.

- APL kann Listen verarbeiten. Jedes Element eines Textes ist einzeln ansprechbar. Mit

  'LISTE'[3]

  erhalte ich ein S. Will ich wissen, wo das erste T auftaucht, frage ich:

  'LISTE' ⍳ 'T'

  Antwort: 4. – Ist ein D enthalten?

  'D' ∊ 'LISTE'

  Antwort: 0. – Logisch.

- Begriffe lassen sich in einer Matrix speichern, suchen, verändern und ersetzen. Die Liste kann sortiert, erweitert, gekürzt werden. Wahlweise können einzelne Zeilen, Spalten, oder Elemente adressiert werden.

- APL kann Daten zu ausführbarem Code machen (und umgekehrt). Listeneintrag Nr 13 enthalte eine Rechenvorschrift **'A×B'** (als editierbaren Text). Mit

  ⍎ Liste[13;]

  wird die Anweisung ausgeführt. Ersetzt man die Vorschrift durch eine andere,

  Liste[13;] ← 'A÷B'

  wird bei der nächsten "Aktivierung" (⍎) eine Division anstelle der Multiplikation ausgeführt. Solche Operationen können geschachtelte Aufrufe enthalten. Besser gesagt: Jeder korrekte APL-Ausdruck wird angenommen.

- Jede selbstgeschriebene Funktion kann mit der Systemfunktion ⎕CR zu Text umgeformt werden, der beliebig vom Programm manipulierbar ist. Mit ⎕FX wird wieder eine Funktion erzeugt, die sofort ausführbar ist. Das heißt: APL-Programme können sich zur Laufzeit selbst verändern und erweitern.

nochmal in die Schule? Soll der doch umdenken.

Soweit die leidige Mathematik. Viele Funktionen gelten gleichermaßen für Zahlen wie für Text: Das Sortieren (auf- und abwärts), das Suchen von Zahlen oder Zeichen („Index"), das Vergleichen, die Abfrage („Element von"), das Umwandeln von Daten („Encode" und „Decode") – damit lassen sich Zahlen in unterschiedliche Zahlensysteme „übersetzen" (z.B. hexadezimal, oktal...) oder Texte „verschlüsseln".

Dann gibt es eine Reihe von Befehlen, um Datenstrukturen zu manipulieren: Rotieren, spiegeln, um einen bestimmten Betrag drehen, in eine gewünschte Richtung... Alles hochinteressant, wenn man mal mit mehr als zwei Dimensionen zu tun hat. Man kann eine Anzahl von Zeichen entnehmen oder entfernen, von „vorn" oder von „hinten" (wo, bitte, hat eine 6-dimensionale Matrix ihr Hinterteil?). In jeder Struktur läßt sich zudem jeder einzelne Wert einzeln ansprechen (per „Indizierung").

Na gut. Die Aufzählung ist lange nicht komplett. Es fehlen etwa die „Operatoren", die so etwas wie „Funktionen der Funktionen" darstellen – das sind Befehle, die als Argument(e) ein oder zwei Funktionen bekommen, die ihrerseits auf Daten operieren. Damit lassen sich beispielsweise Vergleiche auf komplexe Datenstrukturen anwenden, Zeilen in Matrizen suchen oder Funktionstabellen erstellen.

Programmierung

Der Sinn der Auflistung war indes nicht, ein Handbuch zu ersetzen, sondern zu verdeutlichen, daß in APL vieles eingebaut ist, was man in anderen Sprachen erst einbauen oder hinzukaufen muß. Zudem läßt sich erahnen, daß durch die Fähigkeit der Funktionen, auf ganzen Vektoren oder Matrizen zu operieren, viele Loops überflüssig werden, und mit ihnen der ganze Programmieraufwand für die notwendigen Kontrollstrukturen. Verstehen Sie, warum APL-Routinen (pardon: „Funktionen") so kurz sein können?

Die Knappheit der Notation ist natürlich keine Tugend „an sich". Sicher ließe sich viel Papier sparen, wenn man die Übersicht über ein Programm auch am Bildschirm behält, aber im Zeitalter des Waldsterbens kommt es auf das bißchen Holz wohl nicht an, oder? Schlimmer wäre es schon, wenn die Computerzeitschriften die APL-Notation verwenden würden: Womit sollten sie die vielen eingesparten Seiten füllen? Und wie käme der Leser noch zu dem unvergleichlichen Genuß, seitenlange Listings ohne Tippfehler in seinen ST zu hacken? Genug gelästert. Das Wesentliche ist doch, bei der Entwicklung und Wartung komplexer Programme die Übersicht zu behalten. Dieses Ziel unterstützt APL vorbildlich.

Üblicherweise wird bei der Programmierung zweigleisig verfahren:

- Ausgehend von der Grobstruktur (z. B. durch ein Flußdiagramm dargestellt) arbeitet man sich allmählich zu den Details vor („Top Down"-Verfahren).
- Man schreibt einige Unterfunktionen, um sie in der Folge zu größeren Einheiten zusammenzufassen („Bottom Up").

APL unterstützt beide Verfahren. Einzelne Funktionen (z. B. Datumsberechnung, Mittelwertbildung, Eingabe-Editor) werden separat gebaut und getestet. Da sie mit derselben Syntax verwendet werden wie die eingebauten APL-Befehle, erweitert man damit den Sprachumfang nach seinen Bedürfnissen. Das Ablaufschema eines Programms überträgt man in eine Oberfunktion (wobei in den meisten Fällen eine direkte Umsetzung des Flußdiagramms möglich ist!); die einzelnen Bearbeitungsschritte werden mit Namen in diese Funktion eingetragen. Nach und nach werden diese Namen dann mit Inhalt gefüllt, indem man sie zu Funktionen macht, die ihrerseits jede Menge von Unterfunktionen aufrufen können.

„Denkwerkzeug"

Beim Erstellen eines Programms ist das Eintippen der Befehle (das, was man gemeinhin „Programmierung" nennt) meist nicht einmal die halbe Miete. Viel mehr Denk-Arbeit ist beim Umsetzen einer Idee in computergerechte Algorithmen gefragt. Abgesehen davon, daß APL durch seine Fähigkeit, auf Strukturen zu arbeiten, viele Umsetzungsschritte (auf Schleifen) überflüssig macht, ergibt sich ein weiterer Vorteil: Ursprünglich ist APL ja als „Beschreibungssprache", als Notation, entwickelt worden. Das heißt, Sie können eine gut durchdachte Idee direkt in APL beschreiben. Da ein Rechner Ihre Beschreibung „versteht", kann er sofort etwas damit anfangen. Die Zeitersparnis ist enorm: In der Industrie wird mit einer dreifach höheren Produktivität kalkuliert.

Um die Datenbehandlung muß man sich nicht allzusehr kümmern. Dank der Fähigkeit von APL, mit mehrdimensionalen Strukturen umgehen zu können, läßt sich etwa eine Adressdatei ganz einfach als dreidimensionales Gebilde aufbauen: Die (zweidimensionalen) „Karteikarten" werden in die dritte Dimension einsortiert. Damit ist der Karteikasten schon fertig. Solche mehrdimensionalen Strukturen machen die Sache nicht komplizierter, sondern einfacher: Man kann bei der Programmierung seine (menschliche) Vorstellung von Datenformaten („Karteikasten") beibehalten. Natürlich lassen sich Variablen auch lokal machen, und zwar auf jeder Ebene der Programmhierarchie.

All dies gibt es natürlich nicht umsonst. APL ist von Natur aus eine Interpretersprache. Das macht es ungeeignet für massenhaft verkaufte Standardprogramme. Allerdings: Das Vorurteil, ein Interpreter müsse langsam sein, trifft hier nicht. Jede ernsthafte Implementierung auf Microcomputern ist in Assembler geschrieben; Funktionen werden zudem in einer Art präcompilierter Form gehalten. Und:

"Datenbank"-Abfrage

Die Suche von Einträgen in Listen nach bestimmten Kriterien ist ein häufig benötigter Vorgang. In APL läßt sich ein relativ einfacher Fall "im Vorbeigehen" lösen:
Es gibt eine Liste, die (zunächst mal) nur aus Schlüsseln besteht. Nennen wir sie "LISTE".

```
ABC5
DEF2
GHI3
XYZ5
```

Diese Matrix kann beliebig breit/lang sein. Es soll nach jeweils einem Schlüssel (hier: einem Zeichen) in jeweils einer Spalte gesucht werden. Eine erste APL-Funktion könnte so aussehen:

```
∇ R← Spalte ABFRAGE Schlüssel ; LOG
  R←LISTE[;Spalte]     ⍝ Spalte extrahieren
  LOG ←R ∊ Schlüssel   ⍝ logischer Vektor
  R←LOG/⍳⍴R            ⍝ Zeilennummern
```

In R (dem Resultat) werden die Nummern der Zeilen von LISTE übergeben, in deren n-ter Spalte der Schlüssel vorkommt. Also:

```
    4 ABFRAGE '5'      ⍝ Funktionsaufruf
1 4                    ⍝ Resultat
```

Jetzt hängen wir an die LISTE mal paar Daten an:

```
ABC5 Astrid, Bildhübsch, Coburg
DEF2 Dieter, Ekel, Frankfurt
GHI3 Gert, High, Indien
XYZ5 Zimmermann, Aktenzeichen, ungelöst
```

Die LISTE enthält all' Ihre Schuldner, Spalte 4 den Betrag in (nicht konvertierbaren) Rubel. Mit einer besseren ABFRAGE wollen Sie wissen, wer von Ihren Schuldnern ein Ekel ist:

```
    2 ABFRAGE 'E'
Dieter, Ekel, Frankfurt
```

Aha! Und wer schuldet Ihnen jeweils 5 Rubel?

```
    4 ABFRAGE '5'
Astrid, Bildhübsch, Coburg
Zimmermann, Aktenzeichen, ungelöst
```

Die passende APL-Funktion sieht dann so aus:

```
∇ R← Spalte ABFRAGE Schlüssel
  R←0 4↓(LISTE[;Spalte] ∊ Schlüssel)⌿LISTE
```

Das wäre zugleich die "Kompaktversion". Die lokale Variable LOG ist nicht mehr nötig, da Zwischenergebnisse direkt weitergereicht werden. Der Klammerausdruck erzeugt den logischen Vektor (im letzten Beispiel: 1 0 0 1), mit dem die LISTE vertikal "komprimiert" wird. Es bleiben also die Zeilen 1 und 4 übrig. Davon werden die ersten 0 Zeilen, 4 Spalten entfernt, so daß kein Schlüsselfeld angezeigt wird. Passen Sie nur auf, daß der Zimmermann Ihnen nicht durch die Lappen geht!

APL arbeitet ja ganze Strukturen auf einen Schlag ab, so daß die bei einem Von-Neumann-Rechner immer nötigen Schleifchen auf der Maschinensprache-Ebene abgewickelt werden – der Interpreter muß die Loop-Verwaltung nicht immer wieder durchkauen. Durch diese Vorteile ist APL ohne weiteres konkurrenzfähig.

Anwendung

Ganz klar, daß ein Interpreter in der Entwicklungsphase eines Programmes von Vorteil ist. Bei einem Fehler hält APL die Ausführung an und gibt eine Meldung aus, wobei die fehlerhafte Stelle in einer Programmzeile genau markiert wird. Der Entwickler kann nun ganz gemütlich „per Hand" herumprobieren, sich Variablen anzeigen lassen, Werte verändern. Anschließend kann das Programm gleich weitermachen.

Anders als manche andere Sprache kennt APL keinen grundsätzlichen Unterschied zwischen dem „Handbetrieb" und einem Programm. Im sogenannten „Tischrechner-Modus" kann man also beliebige Operationen ausführen, die aus APL-Befehlen und eigenen Funktionen bestehen können. Dieser Modus wird oft von Entwicklungsingenieuren benutzt, die mit ihrem wissenschaftlichen Taschenrechner nicht zufrieden sind, da er bei linearen Gleichungssystemen nicht mehr mag. Auch für Ausbildungszwecke ist dieser Modus ideal: Mathematische Formeln lassen sich leicht ausprobieren, die Ergebnisse sind direkt verfügbar. Komplexe Vorgänge kann man Schritt für Schritt überprüfen. Der Umgang mit dem Rechner ist dabei genauso simpel wie bei einem Taschenrechner: Wenn Sie „1+1" eingeben, antwortet APL mit einer „2". Ich habe gehört, das sei nicht selbstverständlich? Warum sollte man denn mit einer zusätzlichen „PRINT"-Anweisung darum betteln müssen, das Ergebnis zu erfahren? Verstehen Sie das?

Für das Erlernen der Sprache ist der Tischrechner-Modus ideal: Alles läßt sich direkt ausprobieren. Zudem hat sich gezeigt, daß gerade die Computer-Neulinge keinerlei Probleme im Umgang mit APL haben, da sie mit ihrem gesunden Menschenverstand viel ungezwungener an diese Sprache herangehen als etwa ein Assembler-Crack, der die „Denkweise" des Rechners schon voll verinnerlicht hat und bei jedem Befehl schon überlegt, was sich der sensible Computer dabei wohl „denkt". Soviel übertriebene Rücksicht ist aber bei APL fehl am Platze. Der Rechner soll kein Pflegefall, sondern ein Werkzeug sein. Daß er diesen Zweck mit Hilfe von APL mit erstaunlicher Eleganz erfüllt, ist für kommerzielle Nutzer unbestritten. Und angesichts der neuen, leistungsfähigen Rechnergeneration, wie sie der ST verkörpert, gibt es keinen Grund mehr, auf ein ebenso leistungsfähiges „Denkwerkzeug" zu verzichten.

Und die Zukunft von APL? Es ist eine Frage der Zeit, bis die heutigen Von-Neumann-Rechner durch Parallelarchitekturen verdrängt werden. Dieser Trend käme APL entgegen: Schon jetzt schließt IBM ihre Vektor-Rechner an APL an, **ohne** eigene Kommandosprache. Diese Number-Cruncher knabbern sich so genüßlich (und parallel) durch die APL-Matrizen, daß es eine Freude ist. Sicher, da wird das Taschengeld schnell knapp, aber wie wär's mit einem Parallelrechner aus vielen 68000-CPUs? Daran wird an der Universität Saarbrücken gebaut. Muttersprache: APL. Aber auch jetzt schon wird APL gebraucht: Wegen der damit möglichen kurzen Entwicklungszeiten für flexible, anpassungsfähige Software. **Darum** vor allem rangiert APL in der Hitliste schon auf Platz 3, nach den „Sauriern" Cobol und Fortran.

Wollen Sie wissen, was APL eigentlich heißt? „A Programming Language" – eine Programmiersprache also. Das stimmt. Aber was für eine!

Wolfgang Thomas

APL-"Icons"

Das erste wichtige Zeichen ist die "Zuweisung" (←). Mit ihr erhalten Variablen einen Inhalt:

```
      C←2×3
```

In C steht jetzt der Wert 6. Damit läßt sich rechnen:

```
      C*99
1.088864373E77
```

Die neunundneunzigste Potenz von 6. Bauen wir eine Matrix:

```
      Mat←3 4⍴⍳12
```

Das Zeichen '⍳' bewirkt hier die Bildung der Zahlenreihe von 1 bis 12. Der Ausdruck '3 4⍴' bildet eine Matrix mit 3 Zeilen, 4 Spalten:

```
      MAT
1  2  3  4
5  6  7  8
9 10 11 12
```

Mit demselben Zeichen wird die Dimension abgefragt:

```
      ⍴MAT
3 4
```

Ein paar Dreher wären auch nicht schlecht, oder?

```
   ⌽MAT        ¯1⌽MAT        ⊖MAT        ⍉MAT
 4  3  2 1    4 1  2  3    9 10 11 12    1 5  9
 8  7  6 5    8 5  6  7    5  6  7  8    2 6 10
12 11 10 9   12 9 10 11    1  2  3  4    3 7 11
                                         4 8 12
```

Das Hochminus '¯' bezeichnet negative Zahlen, mit dem Minuszeichen '–' ist die Rechenoperation Subtrahieren belegt. Übrigens: Statt der Zahlen kann man auch Zeichen verwenden; anstelle einer zweidimensionalen Matrix geht auch eine mit 7 Dimensionen – oder ein Vektor. Wie's beliebt.

Logische Vektoren spielen in APL eine große Rolle, kann man mit ihnen doch hervorragend Datenstrukturen manipulieren – hier einen Vektor:

```
      0 1 0 0 1 0 1 0 0 0 0/'ST-COMPUTER'
TOP
```

Der Schrägstrich bedeutet dabei: "Komprimieren". Er kann auch als Operator verwendet werden, d.h. mit einer APL-Funktion als (linkem) Argument.

```
      +/4 5 ¯1 3 9 7    ⍝ Aufaddieren des Vektors
27
      ⌈/4 5 ¯1 3 9 7    ⍝ Maximum des Vektors
9
      ⌊/4 5 ¯1 3 9 7    ⍝ Minimalwert
¯1
```

Und noch ein paar Pfeile auf die ST abschießen?

```
      6↓'ST-COMPUTER'
PUTER
      ¯5↑'ST-COMPUTER'
PUTER
```

... noch ein Tier? – Bitte sehr:

```
      'ST-COMPUTER'[1 2 8 9 10]
STUTE
```

Ein "Fenster" dient der Ein- und Ausgabe (I/O):

```
      R←⍞               ⍝ hole Text vom Bildschirm

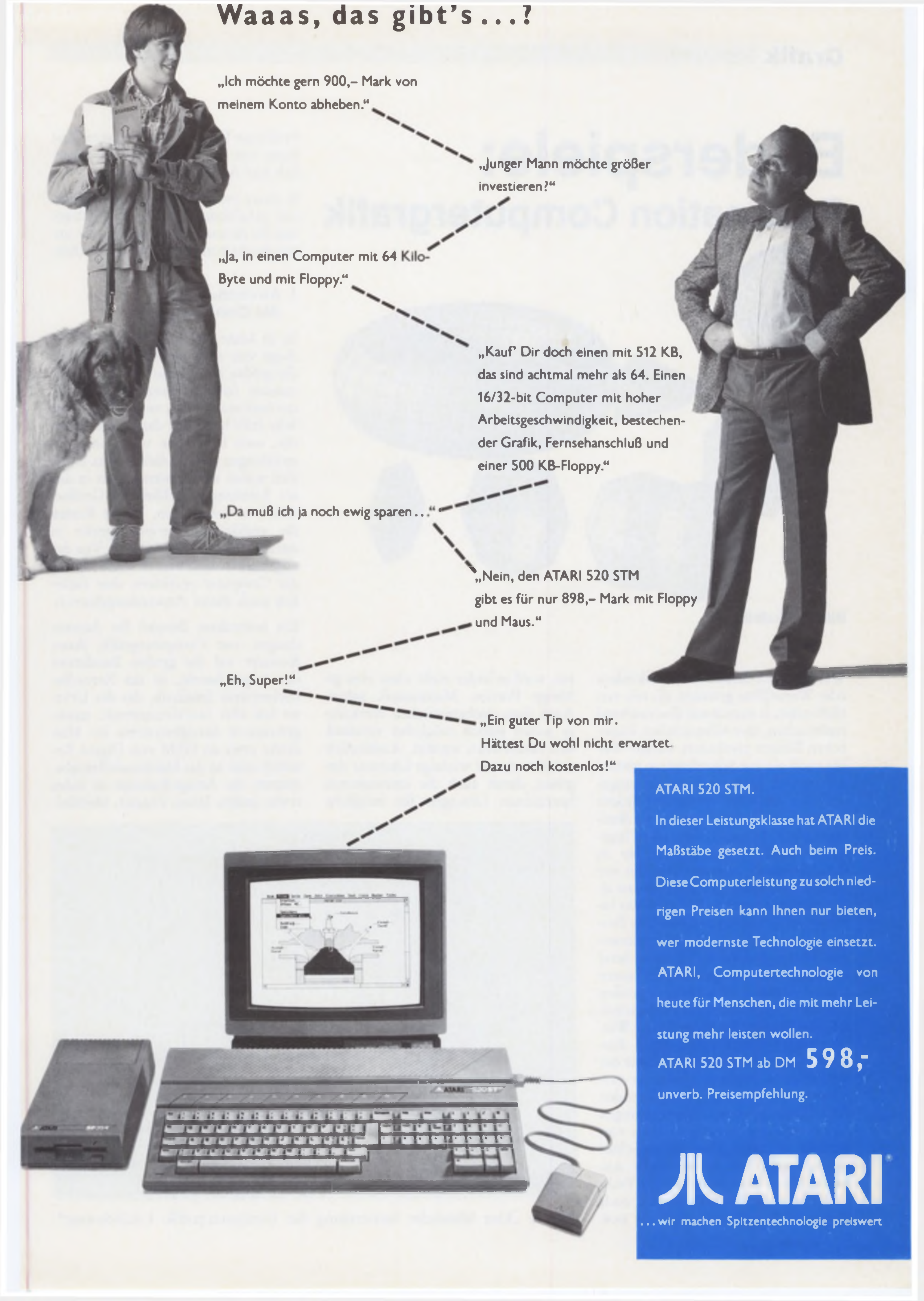
      6+⎕               ⍝ Eingabeanforderung
⎕:                      ⍝ der 'Prompt'
      4×2               ⍝ die Eingabe (mit RETURN)
14                      ⍝ '6' plus Eingabe
R←⍞←'Hallo'             ⍝ ... Text gleichzeitig an R
Hallo                   ⍝       und auf den Schirm
```

Na, es reicht zunächst mal. Die ST braucht ihre Seiten noch für endlose Basic- und C-Listings. Immerhin ist das Prinzip der Sprache APL etwas deutlicher geworden. Möchten Sie sich immer noch die Finger wundtippen? – I like Icons!

APL auf dem ST

Dieser APL-Bericht ist nicht als theoretische Trockenübung gedacht. Wer will, kann etwa die Beispiele auf seinem ST nachvollziehen.
Das APL/68000-ST, mit dem sie auch geschrieben wurden, läuft auf jedem ST ohne Einschränkungen. Ein ausführlicher Test folgt in der nächsten Ausgabe der ST-Computer.

Bezugsquelle: gdat mbH
Stapelbrede 39
48 Bielefeld 1


Waaas, das gibt's ...?
„Ich möchte gern 900,– Mark von meinem Konto abheben."
„Junger Mann möchte größer investieren?"
„Ja, in einen Computer mit 64 Kilo-Byte und mit Floppy."
„Kauf' Dir doch einen mit 512 KB, das sind achtmal mehr als 64. Einen 16/32-bit Computer mit hoher Arbeitsgeschwindigkeit, bestechender Grafik, Fernsehanschluß und einer 500 KB-Floppy."
„Da muß ich ja noch ewig sparen..."
„Nein, den ATARI 520 STM gibt es für nur 898,– Mark mit Floppy und Maus."
„Eh, Super!"
„Ein guter Tip von mir. Hättest Du wohl nicht erwartet. Dazu noch kostenlos!"
ATARI 520 STM.
In dieser Leistungsklasse hat ATARI die Maßstäbe gesetzt. Auch beim Preis. Diese Computerleistung zu solch niedrigen Preisen kann Ihnen nur bieten, wer modernste Technologie einsetzt. ATARI, Computertechnologie von heute für Menschen, die mit mehr Leistung mehr leisten wollen.
ATARI 520 STM ab DM 598,-
unverb. Preisempfehlung.
ATARI
...wir machen Spitzentechnologie preiswert

Bilderspiele:
Faszination Computergrafik

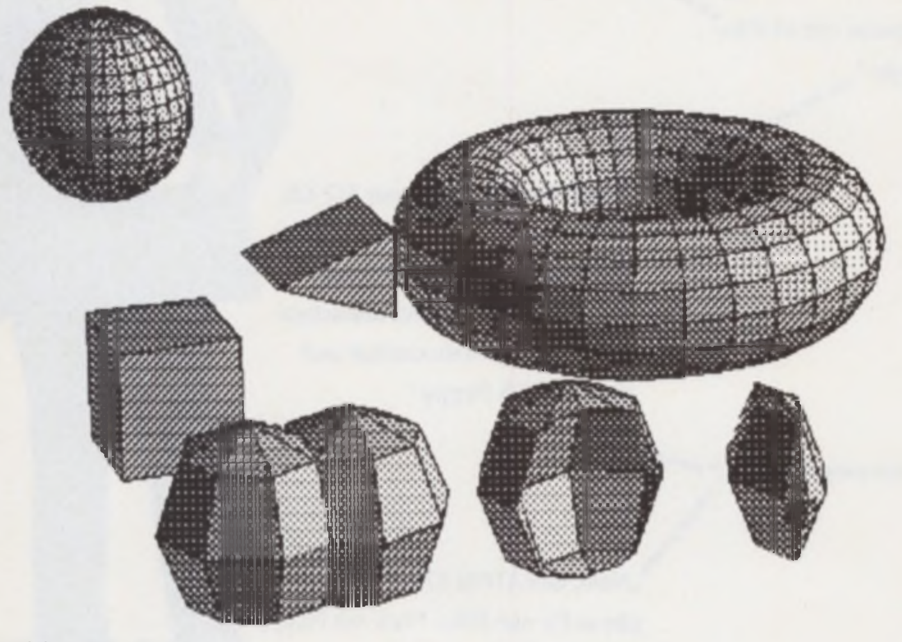


Bild 1: Titelbild

Wer hat noch nicht über Musikvideos oder Werbefilme gestaunt, die mit verblüffenden, ja manchmal überraschend realistischen, aber offensichtlich künstlichen Bildern produziert waren? Wer hat noch nie vor Schaufenstern gestanden und die Demografiken der ausgestellten Computer betrachtet? Jedem Manager ist Computergrafik als Business-Grafik bekannt, fast allen Ingenieuren und Architekten ist sie in CAD-Systemen vertraut. Die Flut der erhältlichen Malprogramme für fast alle Computer beweist, wie groß das Interesse an Grafik ist und wieviele Programmierer sich damit auseinandersetzen. Darum wollen wir in dieser Serie versuchen, unseren geschätzten Lesern einen Überblick über die drei großen Ws der Computergrafik zu verschaffen: Das **Wofür**, das **Was** und das **Wie**. Wir hoffen, Ihnen damit einen Einblick in eines der aktivsten Gebiete der modernen Computertechnik zu verschaffen. Die kaum zu überblickenden Aktivitäten in dieser Sparte bringen leider auch eine ungeheure Fülle von Material mit sich, über das berichtet werden soll. Und da wir nicht ausschließlich an der Oberfläche der Techniken bleiben, sondern Ihnen auch einiges an Knowhow vermitteln wollen, wird es leider nicht ohne eine gehörige Portion Mathematik gehen. Auch diese mathematischen Werkzeuge sollen jedoch möglichst verständlich beschrieben werden. Ausführlich werden wir auf wichtige Literatur eingehen, damit auch die interessierten Spezialisten Lösungen für mögliche Probleme finden – denn unsere Serie kann von dem riesigen Gebiet natürlich nur Ausschnitte zeigen.

In dieser ersten Folge wird es zunächst um verschiedene Anwendungsgebiete und die daraus resultierenden, sehr unterschiedlichen Grafiksysteme gehen.

I. Anwendungen der Computergrafik

Es ist keine neue Erkenntnis, daß das Auge von allen menschlichen Sinnen die größte Bandbreite besitzt, also die meisten Informationen innerhalb einer bestimmten Zeit verarbeiten kann. Sehr früh begannen darum die Versuche, auch Ergebnisse von Computerrechnungen grafisch darzustellen. Lange Zeit waren alle Anwendungen in dieser Richtung ausschließlich Großanwendern vorbehalten, da die Kosten für grafikfähige Peripheriegeräte in astronomischen Höhen lagen. Von der allgemeinen technischen Entwicklung der Computer profitierte aber natürlich auch dieser Anwendungsbereich.

Ein besonderes Beispiel für Anwendungen von Computergrafik, deren Konzept auf der großen Bandbreite des Auges beruht, ist das Xerox-Betriebssystem Smalltalk, das der Urvater fast aller mausorientierten, menügesteuerten Betriebssysteme ist. Man denke etwa an GEM von Digital Research oder an das MacIntosh-Betriebssystem; das Amiga-Konzept ist indes etwas anders. Icons, Fenster, Menülei-

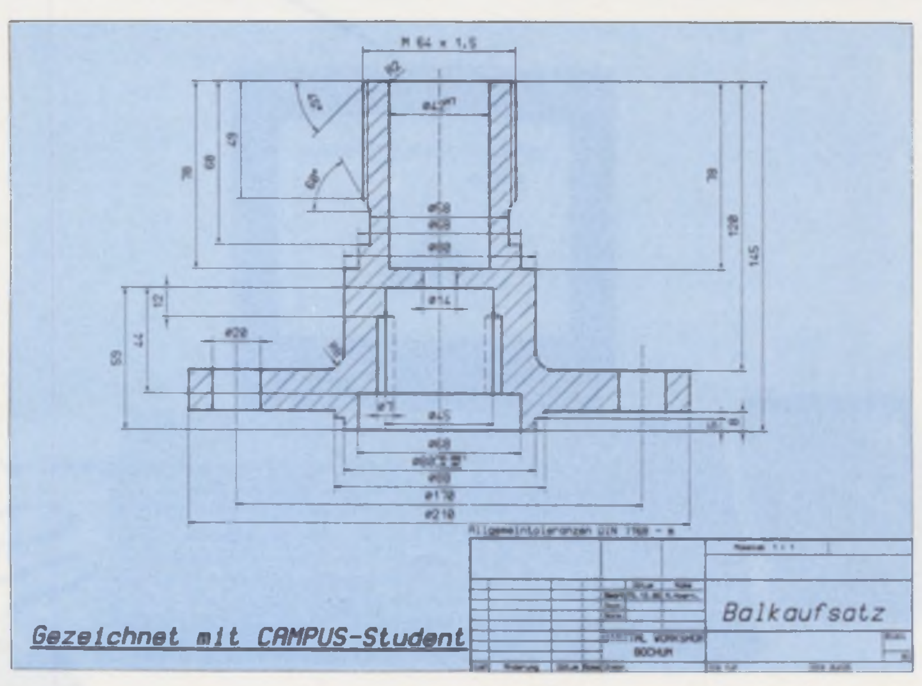
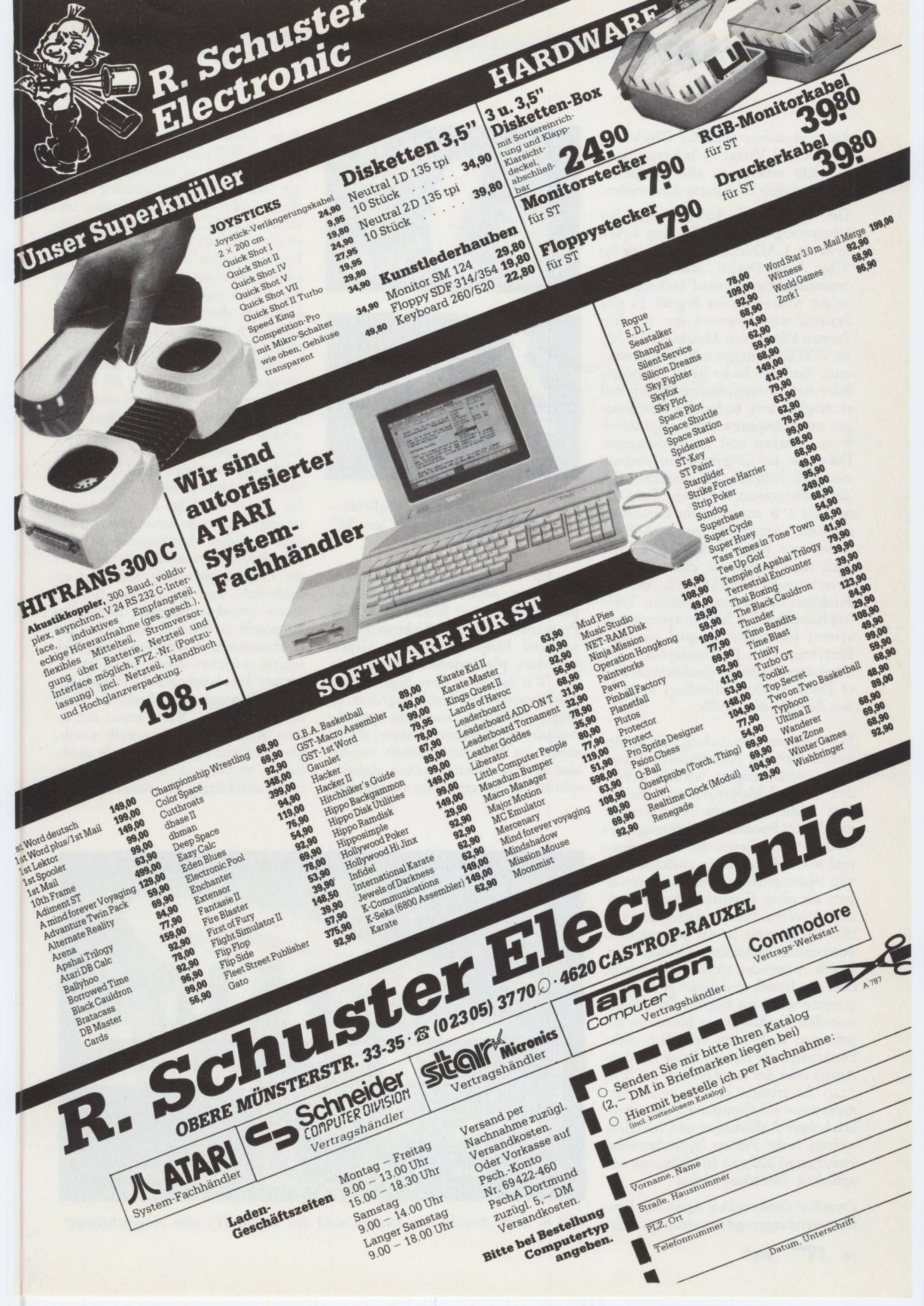


Bild 2: „Eine klassische Anwendung der Computergrafik: CAD-System“



R. Schuster Electronic
HARDWARE
Unser Superknüller
JOYSTICKS
Joystick-Verlängerungskabel 2 × 200 cm 24,90
Quick Shot I 9,95
Quick Shot IV 19,80
Quick Shot V 24,90
Quick Shot VII 27,95
Quick Shot II Turbo 19,95
Speed King 29,80
Competition-Pro 34,90
mit Mikro-Schalter 34,90
wie oben, Gehäuse transparent 49,80
Disketten 3,5"
Neutral 1 D 135 tpi 10 Stück 34,90
Neutral 2 D 135 tpi 10 Stück 39,80
Kunstlederhauben
Monitor SM 124 29,80
Floppy SDF 314/354 19,80
Keyboard 260/520 22,80
3 u. 3,5" Disketten-Box
mit Sortiereinrichtung und Klappdeckel, Klarsichtdeckel, abschließbar
24.90
Monitorstecker für ST 7.90
Floppystecker für ST 7.90
RGB-Monitorkabel für ST 39.80
Druckerkabel für ST 39.80
Wir sind autorisierter ATARI System-Fachhändler
HITRANS 300 C
Akustikkoppler, 300 Baud, vollduplex, asynchron, V 24 RS 232 C-Interface, induktives Empfangsteil, eckige Höreraufnahme (ges. gesch.), flexibles Mittelteil, Stromversorgung über Batterie, Netzteil und Interface möglich, FTZ.-Nr. (Postzulassung) incl. Netzteil, Handbuch und Hochglanzverpackung.
198,—
SOFTWARE FÜR ST
1st Word deutsch 149,00
1st Word plus/1st Mail 199,00
1st Lektor 149,00
1st Spooler 99,00
1st Mail 99,00
10th Frame 63,90
Adiment ST 499,00
A mind forever Voyaging 129,00
Advanture Twin Pack 59,90
Alternate Reality 69,90
Arena 84,90
Apshai Trilogy 77,90
Atari DB Calc 159,00
Ballyhoo 92,90
Borrowed Time 78,00
Black Cauldron 92,90
Bratacass 96,90
DB Master 99,00
Cards 56,90
Championship Wrestling 68,90
Color Space 69,90
Cutthroats 92,90
dbase II 348,00
dbman 399,00
Deep Space 94,90
Eazy Calc 119,00
Eden Blues 76,90
Electronic Pool 54,90
Enchanter 92,90
Extensor 69,90
Fantasie II 78,00
Fire Blaster 53,90
First of Fury 39,90
Flight Simulator II 148,50
Flip Flop 39,90
Flip Side 57,90
Fleet Street Publisher 375,90
Gato 92,90
G.B.A. Basketball 89,00
GST-Macro Assembler 149,00
GST-1st Word 99,00
Gaunlet 79,95
Hacker 78,00
Hacker II 67,90
Hitchhiker's Guide 89,00
Hippo Backgammon 99,00
Hippo Disk Utilities 149,00
Hippo Ramdisk 99,00
Hipposimple 149,00
Hollywood Poker 29,90
Hollywood Hi Jinx 92,90
Infidel 92,90
International Karate 62,90
Jewels of Darkness 62,90
K-Communications 149,00
K-Seka (6800 Assembler) 149,00
Karate 62,90
Karate Kid II 63,90
Karate Master 40,90
Kings Quest II 92,90
Lands of Havoc 56,90
Leaderboard 68,90
Leaderboard ADD-ON T 31,90
Leaderboard Tornament 32,90
Leather Goddes 78,90
Liberator 35,90
Little Computer People 80,90
Macadum Bumper 77,90
Macro Manager 119,00
Major Motion 51,90
MC Emulator 598,00
Mercenary 63,90
Mind forever voyaging 108,90
Mindshadow 80,90
Mission Mouse 69,90
Moonmist 92,90
Mud Pies 56,90
Music Studio 108,90
NET-RAM Disk 49,00
Ninja Mission 29,90
Operation Hongkong 59,90
Paintworks 109,00
Pawn 77,90
Pinball Factory 69,90
Planetfall 92,90
Plutos 41,90
Protector 53,90
Protect 148,00
Pro Sprite Designer 104,90
Psion Chess 77,90
Q-Ball 54,90
Questprobe (Torch, Thing) 69,90
Quiwi 69,90
Realtime Clock (Modul) 104,90
Renegade 29,90
Rogue 78,00
S.D.I. 109,00
Seastalker 92,90
Shanghai 68,90
Silent Service 74,90
Silicon Dreams 62,90
Sky Fighter 59,90
Skyfox 68,90
Sky Plot 149,00
Space Pilot 41,90
Space Shuttle 79,90
Space Station 63,90
Spiderman 62,90
ST-Key 79,90
ST Paint 99,00
Starglider 68,90
Strike Force Harrier 68,90
Strip Poker 49,90
Sundog 95,90
Superbase 249,00
Super Cycle 68,90
Super Huey 54,90
Tass Times in Tone Town 68,90
Tee Up Golf 41,90
Temple of Apshai Trilogy 79,90
Terrestrial Encounter 39,90
Thai Boxing 39,90
The Black Cauldron 89,00
Thunder 123,90
Time Bandits 84,90
Time Blast 29,90
Trinity 108,90
Turbo GT 49,90
Toolkit 99,00
Top Secret 59,90
Two on Two Basketball 68,90
Typhoon 48,90
Ultima II 89,00
Wanderer 68,90
War Zone 69,90
Winter Games 68,90
Wishbringer 92,90
Word Star 3.0 m. Mail Merge 199,00
Witness 92,90
World Games 68,90
Zork I 86,90
R. Schuster Electronic
OBERE MÜNSTERSTR. 33-35 · ☎ (02305) 3770 · 4620 CASTROP-RAUXEL
ATARI System-Fachhändler
Schneider Computer Division Vertragshändler
star Micronics Vertragshändler
Tandon Computer Vertragshändler
Commodore Vertrags-Werkstatt
Laden-Geschäftszeiten
Montag – Freitag 9.00 – 13.00 Uhr 15.00 – 18.30 Uhr
Samstag 9.00 – 14.00 Uhr
Langer Samstag 9.00 – 18.00 Uhr
Versand per Nachnahme zuzügl. Versandkosten. Oder Vorkasse auf Psch.-Konto Nr. 69422-460 PschA Dortmund zuzügl. 5,– DM Versandkosten.
Bitte bei Bestellung Computertyp angeben.
Senden Sie mir bitte Ihren Katalog (2,– DM in Briefmarken liegen bei)
Hiermit bestelle ich per Nachnahme: (incl. kostenlosem Katalog)
Vorname, Name
Straße, Hausnummer
PLZ, Ort
Telefonnummer
Datum, Unterschrift
A 787

sten und andere grafische Objekte – all dies sind Versuche, Informationen in leicht erfaßbarer, also *grafischer* Form auf dem Bildschirm darzustellen.

Die 'klassischen' Anwendungen der Computergrafik sind allerdings CAD-Systeme. CAD ist eine Abkürzung für 'Chip Aided Design', was etwa 'rechnerunterstützter Entwurf' bedeutet. Das ist ein weitreichender Begriff. Es gibt Systeme zum Entwurf der verschiedensten Dinge, von Maschinen bis hin zu VLSI-Schaltungen oder Platinenlayouts. Sinn der Sache ist es, Entwürfe jeder Art zu vereinfachen. Damit soll es möglich sein, technische Zeichnungen maßstabsgetreu mit Computerhilfe zu erstellen und zu modifizieren. Dabei ist der Computer im einfachsten Fall lediglich Zeichenhilfe, also Zeichenbrettersatz. Andere Funktionen, die z. B. automatische Bemaßung erlauben, können hinzukommen. Grosse Systeme können das Gezeichnete auch dreidimensional darstellen und Berechnungen durchführen, die die Funktion der zu entwerfenden Technik betreffen. Ein Schaltungsentwurfssystem kann zum Beispiel vor dem Berechnen des Platinenlayouts eine Schaltungssimulation durchführen. Diese Dinge gehören allerdings nicht in den Bereich der Grafik.

Eine weitere Anwendung, die in der letzten Zeit für professionelle Video- und Grafik-Designer wichtig geworden ist, sind Malprogramme, die auf dem Bildschirm mehr oder weniger gut die traditionellen Werkzeuge des Grafikers simulieren. Im allgemeinen sind solche Programme zusätzlich um sehr leistungsfähige Funktionen erweitert, die ohne Computer kaum realisierbar wären. Haupteinsatzgebiet solcher Programme ist die nachträgliche Bearbeitung digitalisierter Videobilder. Jedem Anwender eines Atari, Amiga oder MacIntosh sind derartige Malprogramme wohl ein Begriff. Die professionellen Ausführungen bieten im allgemeinen auch nicht erheblich mehr Funktionen als Deluxe Paint auf dem Amiga oder der Art Director auf dem Atari. Sie arbeiten aber natürlich normalerweise auf Computern, die erheblich höhere Auflösungen und vor allem weitaus mehr Farben (typischerweise zwischen 65 000 und 16 Millionen verschiedene Farbtöne) bieten.

Zwischen diesen beiden Arten von Grafikanwendungen gibt es einen grundlegenden Unterschied, der deshalb so wichtig ist, weil er die gesamte Computergrafik in zwei Welten teilt. CAD-Programme arbeiten **objektorientiert**, während Malprogramme im Gegensatz dazu **pixelorientiert** funktionieren. Was bedeuten diese Begriffe? Ein objektorientiertes Grafikprogramm erzeugt, gleichgültig was seine spezielle Aufgabe ist, grafische Objekte, die als solche im Speicher gehalten werden und erst in einem zweiten Arbeitsschritt auf ein grafisches Ausgabemedium (siehe Info-Kasten) ausgegeben werden. Will man zum Beispiel einen Kreis zeichnen, so speichert der Rechner intern die Beschreibung dieses Kreises, also seinen Mittelpunkt und seinen Radius. Eine eigene Zeichenroutine zeichnet diesen Kreis. Weil die Beschreibungen aller grafischen Objekte gespeichert sind, ist es kein Problem, Teile eines Bildes zu verändern. Um z. B. den oben erwähnten Kreis zu ändern, braucht man nur entweder seinen Radius oder seinen Mittelpunkt zu verändern, muß dann aber leider das ganze Bild löschen und alle Objekte neu zeichnen. Das Programm weiß ja nicht, welche Pixel auf dem Bildschirm zu dem Objekt gehören, oder welche anderen Objekte von dieser Änderung betroffen sind. Es wäre auch sehr aufwendig, dies festzustellen. Daher wird bei einer Änderung im allgemeinen das ganze Bild neu aufgebaut. Sollen dreidimensionale Objekte konstruiert werden, ist diese Art der Objektspeicherung unbedingt erforderlich, weil das Objekt, um auf einem zweidimensionalen Bildschirm dargestellt zu werden, erst einmal auf mathematischem Wege auf diesen Bildschirm projiziert werden muß. Im Gegensatz dazu bearbeiten die Funktionen eines Malprogramms direkt die Pixel des Bildschirms. Wenn etwas mit einem Malprogramm dargestellt wurde, kann es nachträglich nicht mehr modifiziert werden, es kann also nicht mehr unabhängig von seinem Hinter-





Bild 3 + 4: „Vielfältige Möglichkeiten von Malprogrammen am ST"

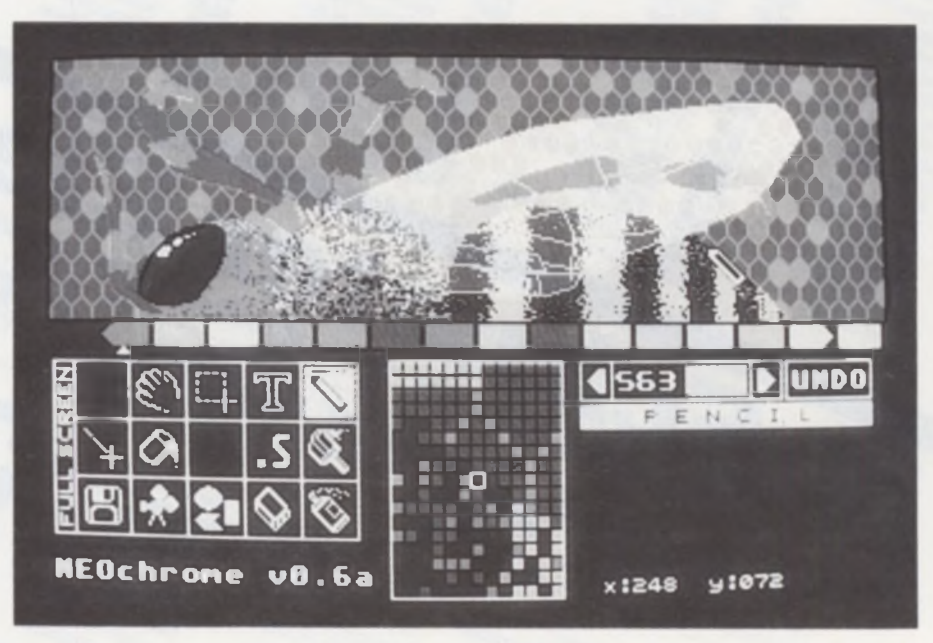


Bild 8: „Die Anzahl der Farben sind auf einem PC sehr eingeschränkt"

grund und seiner näheren Umgebung bearbeitet werden. Das ist wie beim Malen auf einem Stück Papier: Wenn Sie einen Kreis in ein Bild hineinmalen, dann können Sie ihn nur mit einem Radiergummi entfernen – wobei auch der Hintergrund verschwindet. Im Gegensatz dazu entspricht objektorientiertes Arbeiten mehr einer Collagetechnik: Sie fertigen die einzelnen Objekte einzeln an und legen sie in die richtige Anordnung. Jetzt können Sie natürlich einzelne Objekte wieder wegnehmen und verändern, ohne damit die anderen Objekte zu beeinflussen.

Die Anwendung von pixelorientierten Programmen ist daher tatsächlich im wesentlichen auf grafische Arbeit im traditionellen Sinn beschränkt, wenn auch häufig mit zahlreichen zusätzlichen Möglichkeiten. Eine Anwendung ist zum Beispiel die Nachbearbeitung von Videobildern oder von Bildern objektorientierter Systeme. Häufig werden solche Systeme auch zur Titel- und Textgenerierung im Fernsehbereich oder eben tatsächlich als Malprogramm eingesetzt.

In dieser Serie soll es uns aber vor allem um objektorientiertes Arbeiten gehen. Uns soll interessieren, wie man Objekte im Rechner speichert und wie man sie auf dem Bildschirm darstellt. Dann wird die dritte Dimension im Mittelpunkt stehen. Wie bekommt man räumliche Objekte auf den flachen, zweidimensionalen Bildschirm?







Bild 5, 6 + 7: „Animation von Bildern ist auch möglich“

Schließlich werden wir uns mit denjenigen Verfahren befassen, die es ermöglichen, mehr oder weniger natürliche Bilder mit dem Computer zu generieren und darzustellen. Licht, Schatten und Farben werden unsere Themen sein. Wie gesagt, all dies erfordert eine Menge Mathematik. Erwarten Sie daher bitte keine abtippfertigen Anwendungsprogramme. Wir möchten Ihnen die mathematischen Verfahren grob erläutern und kleine Beispielalgorithmen angeben. Außerdem werden wir Literatur vorstellen, die Ihnen bei näherem Interesse weiterhelfen kann. Doch nun weiter zum eigentlichen Thema dieses Kapitels: Wofür – außer die rein technisch orientierten CAD-Anwendungen – braucht man Verfahren, die der Generierung natürlich aussehender Bilder imaginärer Welten dienen, und die dazu auch noch Techniken und Geräte erfordern, die ungeheuer teuer sind?

Denken Sie einmal an alle Arten von Simulatoren. Der Versuch, Kosten und Risiken von Testflügen neuer Flugzeugtypen zu vermeiden, führte zur Entwicklung von Flugsimulatoren, die die Bewegungen eines Cockpits recht gut vortäuschen können und über aufwendige Rechenverfahren auch das Verhalten des geplanten Flugzeuges. Dabei zeigte es sich rasch, daß eine für den Piloten ausreichende Simulation nicht nur künstliche Bewegungen des Cockpits erfordert, sondern auch die Erzeugung einer optischen Reaktion auf die Bewegungen des Flugzeuges, also die halbwegs realistische Simulation einer Landschaft, in der das Flugzeug sich bewegt.

Inzwischen ist man so weit, daß man bei einer simulierten Landung zum Beispiel auf einem Flugzeugträger sogar die Bremsspuren auf dem Deck erkennen kann und die Wellen im umgebenden Ozean oder die Wolken am Himmel erstaunlich realistisch wiedergegeben werden können. Bei solchen Simulationen ist es besonders wichtig, zu beachten, daß die Bilder dabei in Real-Time, für eine halbwegs gute Filmwirkung mit einer Rate von mindestens 20 Bildern pro Sekunde berechnet und dargestellt werden müssen. Eine andere Anwendung sind Filmproduktionen. In dieser Sparte sind auch Musikvideos und Werbefilme enthalten. Hier bieten sich über Computeranimation Spezialeffekte an,

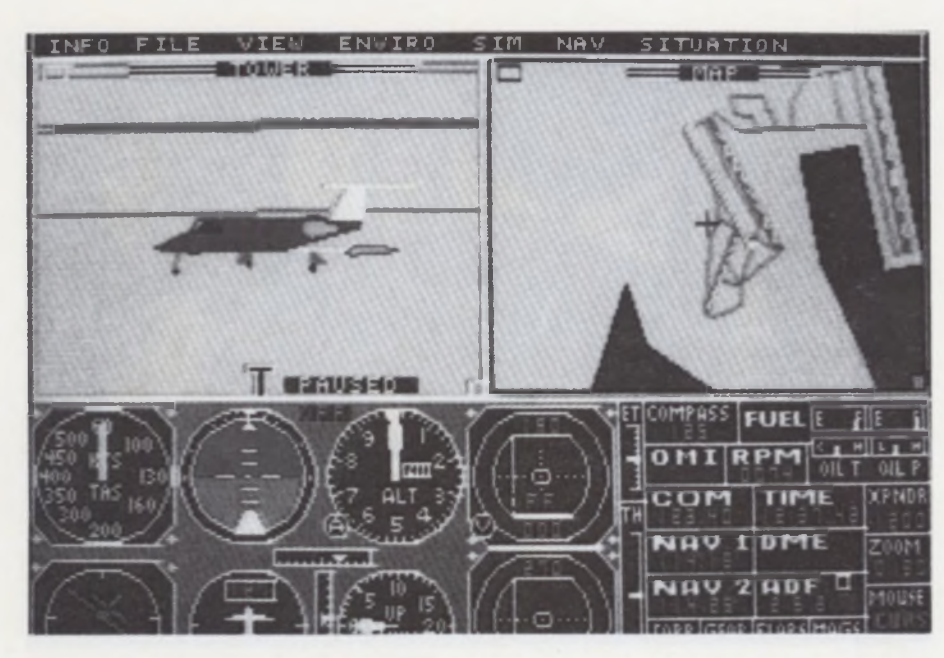


Bild 9: „Ein Flugsimulator am ST“

die auf diese Weise oft billiger zu erstellen sind als mit den traditionellen Techniken. Ein Beispiel für einen solchen Film ist die Walt-Disney-Produktion „Tron", bei der der größte Teil des Films komplett vom Computer gezeichnet wurde. Selbst die Kostüme wurden den Darstellern vom Computer auf den Leib gerechnet. Für Animationsfilme mit Computerunterstützung wurden spezielle Animationssprachen und Programmpakete entwickelt. Viele dieser Systeme simulieren die tatsächliche Arbeit einer Kamera: Der Regisseur kann den Kamerastandpunkt ebenso bestimmen wie Blickrichtung, Bildausschnitt, Objektivbrennweite und Szenenbeleuchtung. Kamerafahrten und Bewegungen von Teilobjekten können programmiert werden. Aus diesen Angaben des Regisseurs, die häufig interaktiv an einem einfachen, vom Computer errechneten Drahtmodell eingegeben werden können, werden dann die einzelnen Bilder errechnet.

Architekten können mit computergenerierten Filmen zukünftigen Auftraggebern eine realistischere Vision eines geplanten Gebäudes geben, als dies mit den üblichen Modellbautechniken möglich ist. Selbst einen Rundgang durch das fertiggestellte Gebäude samt Aussicht kann der Computer berechnen. Natürlich ist so etwas sehr aufwendig und heute noch sehr teuer. Die Kosten für derartige Produktionen fallen aber gemeinsam mit den Computerpreisen rasant.

Weitere wichtige Anwendungen für Computergrafik liegen im Business- und Präsentationsbereich. Statistische Daten lassen sich erheblich einprägsamer in grafisch gestalteter Form präsentieren. Die große Arbeit, mit der das Erstellen solcher Grafiken verbunden ist, kann der Computer übernehmen. Schließlich ist auch das in letzter Zeit vielgerühmte Desktop Publishing eine Form der objektorientierten Grafikanwendung. Auch Buchstaben werden dabei als grafische Objekte aufgefaßt und mit Spezialprogrammen für ihre Gestaltung bearbeitet.

II. Was ist machbar? Die Möglichkeiten der Computergrafik

Auch hier ist es wieder notwendig, sich die verschiedenen Arten der Computergrafik bewußt zu machen. Werfen wir zunächst einen Blick auf zweidimensionale Malprogramme: Um hier die Entwicklung zu sehen, genügt es, auf der einen Seite einfache, aber dennoch zu ihrer Zeit revolutionäre Programme wie MacPaint auf dem MacIntosh zu betrachten. Diese Programme erlauben es, schnell und problemlos einfache geometrische Formen zu zeichnen und Flächen mit Mustern zu füllen. Außerdem ist es möglich, mit einer simulierten Sprühdose oder verschiedenen Pinseln zu arbeiten und auf dem Bildschirm zu zeichnen wie auf einem Block. Kopieren oder Verschieben von Bildausschnitten ist genauso möglich wie die grafische Darstellung von Text. Neuere Programme, wie zum Beispiel der Art Director auf dem Atari, bieten da einiges mehr. Zum Beispiel kann man Bilder oder Bildausschnitte auf alle denkbaren Arten verzerren, drehen oder spiegeln. Andere Programme erlauben auch die Darstellung von Splines, also beliebig verformbaren Kurven. Zusätzliche Features betreffen die Farbdarstellung und Bearbeitung. Besonders bei den professionellen Programmen, die über große Farbauswahl verfügen, sind diese Möglichkeiten zur Farbbearbeitung ausgeprägt. So gibt es zum Beispiel Pinselmodi, in denen sich der Computerpinsel verhält wie ein Pinsel und Ölfar-

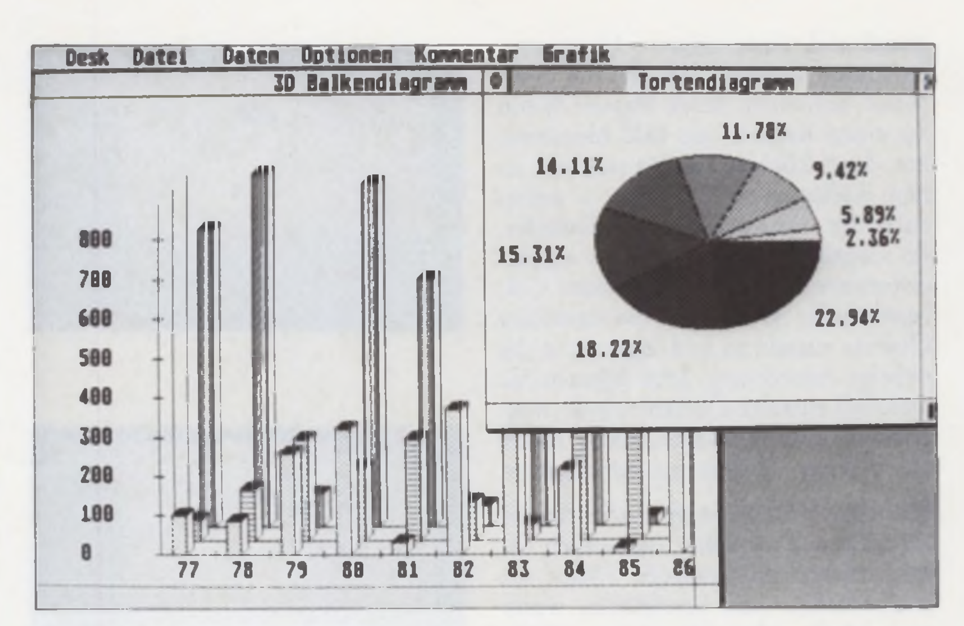


Bild 10: „Balken- und Kuchendiagramme sind an PC's kein Problem"



Bild 11: „Bilder kann man auf alle denkbaren Arten weiterverarbeiten"

ben; man kann auf diese Weise Farben beliebig verschmieren und malen wie auf Leinwand. Im Grunde sind die Möglichkeiten von Malprogrammen aber wohl ziemlich ausgereizt; die Simulation der klassischen Malwerkzeuge ist inzwischen bei den meisten Programmen recht ausgereift, Erweiterungen bei neueren Programmen enthalten meist nur neue Spezialeffekte, die bei der täglichen Arbeit, wie sie auf dem Profi-Sektor vorkommt, wohl selten benötigt werden. Zum Spielen zu Hause werden diese Features aber immer beliebter.

Objektorientierte Zeichenprogramme sind immer stärker 3D-orientiert. Die Zeit, in der sich die Ausgabe der typischen Konstruktionsprogramme im wesentlichen an der Arbeit des technischen Zeichners orientierte, ist im professionellen Bereich längst vorbei. Es werden also nicht nur Pläne von Objekten gezeichnet, der Computer wird auch Werkzeug zur räumlichen Darstellung der konstruierten Objekte und ermöglicht dem Benutzer so eine genauere Vorstellung des Objekts. Dabei ist es wichtig, daß ein Konstruktionsprogramm nicht nur Linien darstellen kann, die die einzelnen Eckpunkte einer Zeichnung verbinden; im Gegenteil, bei der räumlichen Darstellung werden Objekte zumindest als aus Flächen, wenn nicht gar aus Volumen zusammengesetzt betrachtet. Wenn ein Körper als Fläche bekannt ist, können Linien, die durch weiter vorne befindliche Objektteile verdeckt werden, ausgeblendet werden; Volumenmodelle erlauben sogar Schnitte durch den Körper. Mit Hilfe der Darstellung von Objekten als räumliche Flächen oder Volumen wird auch die Erzeugung von Licht- und Schatteneffekten auf den Oberflächen des Objektes möglich. Auf diese Art können recht realistische Bilder der konstruierten Objekte erzeugt werden. Einige der dafür erforderlichen Techniken sollen in weiteren Folgen erläutert werden.

Inzwischen ist die Entwicklung so weit, daß selbst mathematisch extrem schwierig zu beschreibende, völlig unregelmäßige Formen wie Wellen oder Wolken realistisch darstellbar sind. Man kann Oberflächenstrukturen von Gegenständen erzeugen und verändern, selbst die Lichtbrechung in transparenten Materialien kann täuschend echt simuliert werden. Die Fortschritte der Computergrafiker sind enorm. Die Verfahren werden immer komplizierter und besser, in höherem Maße, als es die Geschwindigkeit der für die Bilderzeugung notwendigen Rechner vermuten läßt. Auch die Techniken, die für realistische Bilder verwendet werden, wollen wir in dieser Serie ansprechen und beschreiben. Es wird uns allerdings nicht möglich sein, dieses ungeheuer umfangreiche Thema vollständig abzuhandeln. Deswegen sollen im Schluß-Kapitel dieser Folge für besonders Interessierte schon einmal einige wichtige Bücher zur Einarbeitung vorgestellt werden.

III. Nun zur Sache

Leider ist es uns aus Platzgründen unmöglich, hier auf die absoluten Grundlagen der Matrix- und Vektorrechnung und ihrer graphischen Interpretatation einzugehen. Wer sich jedoch darüber genau informieren möchte, dem seien nachdrücklich die Bücher 'Graphische Datenverarbeitung' oder 'Mathematical Elements for Computer Graphics' empfohlen. Näheres über beide Bücher im nächsten Kapitel.

Als Beispiel stellen wir uns zunächst einen Kubus vor, der irgendwo im Raum herumschwebt. Dieser Kubus läßt sich jetzt beschreiben:

1. Durch seine Eckpunkte.
2. Durch Nennung der Eckpunkte, die durch Linien verbunden sind.
3. Falls der Kubus massiv sein soll, müssen auch noch die Linien oder

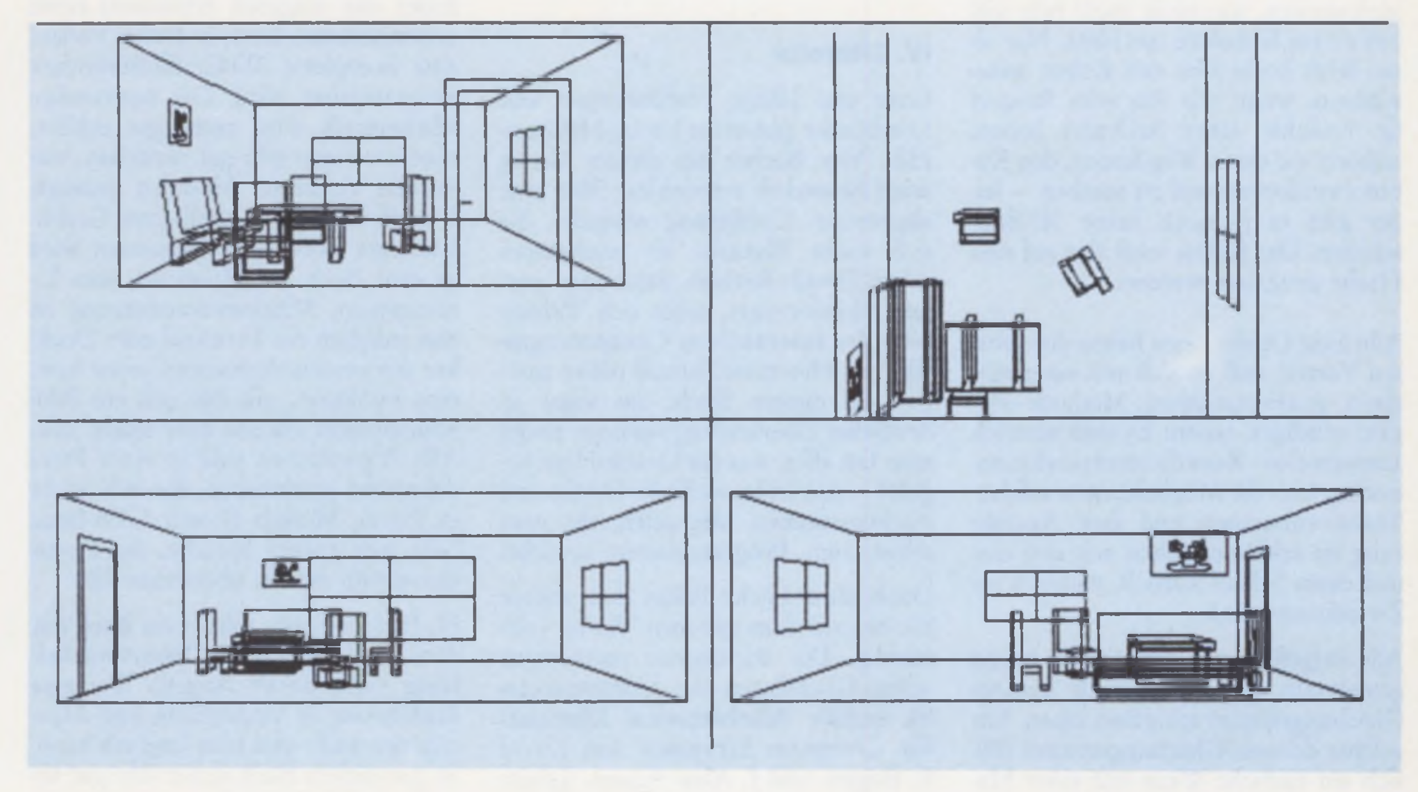


Bild 12: „Perspektivische Darstellung eines Raums“

Ecken, die die Flächen umschließen, genannt werden.
4. Falls die Flächen eine Struktur haben oder auf andere Weise realistisch aussehen sollen, müssen noch andere Daten zusätzlich zur Verfügung stehen, z. B. Brechungsindizes usw. der Oberfläche.

Im Moment sollen uns allerdings nur die Eckpunkte interessieren. Denn es ist offensichtlich, daß nur die Position dieser Punkte im Raum Aussehen und Lage des Kubus bestimmt (alle anderen Angaben sind auf die Position dieser Punkte bezogen, eine Linie kann ja in der Art 'von Punkt A nach Punkt B' beschrieben werden.

Was könnte man mit einem Kubus im Raum, also mit seinen Eckpunkten, tun? Zuerst einmal könnte man den ganzen Kubus durch den Raum bewegen, entweder entlang einer Linie oder auch um einen Punkt rotierend. Weiterhin könnte man den Kubus vergrößern und verkleinern, entweder insgesamt oder teilweise. 'Teilweise' soll in diesem Zusammenhang heißen, daß nur ein Teil der Koordinaten der Punkte verändert wird. Aus einem Kubus könnte also ein Schukarton werden oder etwas noch viel flacheres. Zuletzt könnte man natürlich noch jeden Punkt des Kubus einzeln bearbeiten. Dabei würde natürlich etwas herauskommen, was nicht mehr sehr viel mit einem Kubus zu tun hätte. Nur eines fehlt noch: Um den Kubus anzuschauen, wenn wir ihn zum Beispiel im Speicher eines Rechners haben, müssen wir einen Weg finden, den Kubus zweidimensional zu machen – leider gibt es ja noch keine 3D-Bildschirme. Der Kubus muß also auf eine Fläche projiziert werden.

Alle diese Operationen haben den grossen Vorteil, daß sie sich mit einer einzigen mathematischen Methode elegant erledigen lassen: Es sind nämlich ausnahmslos Koordinatentransformationen. Um die Möglichkeiten solcher Transformationen und ihre Ausführung zu erklären, gehen wir erst einmal einen Schritt zurück, nämlich ins Zweidimensionale.

Alle aufgeführten Operationen haben gemeinsam, daß sie sich als lineares Gleichungssystem schreiben lassen. Ein solches lineares Gleichungssystem läßt sich auf einfache Weise mit einer Matrix schreiben. In dieser Matrix haben nun verschiedene Elemente ganz einfache geometrische Auswirkungen. Für jede der oben angeführten Operationen gibt es ein Element in der Matrix, das genau die gewünschten Auswirkungen hat, und es ist möglich, mit dem Produkt mehrerer Matrizen beliebig komplexe Transformationen in einem Rutsch durchzuführen. Mehr soll in dieser ersten Folge noch nicht verraten werden; in der nächsten Folge werden wir solche Matrizen genauer betrachten.

So vielseitig eine Matrix ist, es gibt auch Probleme der Computergrafik, die nicht mit einfachen Matrixoperationen gelöst werden können. Denken Sie noch einmal an den Kubus im Raum. Wenn Sie irgendeinen Schuhkarton betrachten, wird Ihnen auffallen, daß Sie nicht alle Flächen und Kanten des Kartons gleichzeitig sehen können. Natürlich nicht, die hinteren Teile werden ja durch die vorderen verdeckt. Dummerweise ist es gar nicht so einfach, festzustellen, welche Teile eines Objektes für einen Betrachter von einer bestimmten Position aus verdeckt sind. Um diese Teile zu finden und bei Bedarf auszublenden, sind weitaus komplexere Verfahren notwendig, die auch nicht mehr rein mathematischer, sondern eher algorithmischer Struktur sind. Aber auch hiermit werden wir uns noch befassen.

IV. Literatur

Gute und billige Einführungen und Lehrbücher gibt es im Verlag McGraw-Hill. Vier Bücher aus diesem Verlag seien besonders empfohlen: Wer eine allgemeine Einführung wünscht, die sich nicht allzusehr an mathematischen Details festhält, dafür aber umfassend informiert, sollte sich **'Prinzipien der interaktiven Computergrafik'** von Newman/Sproull näher ansehen. In diesem Buch, das sogar in deutscher Übersetzung vorliegt, findet man fast alles, was das Grafik-Herz begehrt – nur leider nicht die Details und mathematischen Methoden, die man selbst zum Programmieren braucht.

Doch diese Lücke füllen drei andere Bücher aus dem gleichen Verlag vollständig. Die wichtigsten mathematischen Grundlagen der Computergrafik enthält **'Mathematical Elements for Computer Graphics'** von David F. Rogers und J. Alan Adams. Leider ist dieses sehr empfehlenswerte Buch nur auf Englisch erhältlich. Es enthält aber, relativ verständlich formuliert, alle wichtigen mathematischen Werkzeuge, die für Computergrafik benötigt werden. Wichtig ist dabei vor allem, daß die Darstellung einheitlich und konsequent ist; man muß sich also nicht immer wieder an neue Schreibweisen gewöhnen, was fast unumgänglich ist, wenn man sich den Inhalt dieses Buches aus verschiedenen anderen Darstellungen zusammensuchen muß.

Wie bereits im letzten Abschnitt erwähnt, es gibt eine ganze Reihe von Problemen (sozusagen die höhere Schule der Computergrafik), die algorithmische Lösungsverfahren erfordern. Eine großangelegte Sammlung wichtiger Algorithmen gibt David F. Rogers in seinem zweiten Grafikbuch **'Procedural Elements for Computer Graphics'**. Dieser Band erfordert allerdings schon recht ausführliche Kenntnisse in Vektor- und Matrixrechnung. Jedem, der sich wirklich intensiv mit der Materie beschäftigen will, sei er jedoch empfohlen.
Wer vor allem an praktischen Problemlösungen interessiert ist und von Theorie nur das unbedingt notwendige Mindestmaß wissen möchte, findet in dem Buch **'Computer Graphics – A Programming Approach'** von Steven Harrington (auch dieses Buch ist leider nur Englisch erhältlich) einen systematischen Kurs, in dessen Verlauf eine komplette 3D-Grafik-Bibliothek programmiert wird. Die notwendige Mathematik wird recht gut erklärt, man kann also sehr gut verstehen, was in den einzelnen Modulen passiert. Vor allem ist keine intelligente Grafikhardware notwendig. Begonnen wird in dem Buch mit Routinen zum Linienziehen, Mindestvoraussetzung ist also lediglich ein Terminal oder Drukker mit positionierbarem Cursor bzw. eine Funktion, mit der sich ein Bildschirmpunkt löschen oder setzen läßt. Alle Algorithmen sind in einer Pseudosprache geschrieben, die sich leicht in Pascal, Modula II oder GFA-Basic oder jede andere Sprache, die Prozeduraufrufe zuläßt, übersetzen läßt.

Im Hanser-Verlag gibt es ein Buch mit dem Titel **'Grafische Datenverarbeitung'** (von Ian O. Angell). Wer eine Einführung in Mathematik und Algorithmen sucht und kein Englisch kann, ist mit diesem Buch sicher sehr gut beraten. Die Erklärungen sind gut ver-

ständlich, allerdings sind die Programmbeispiele in Fortran geschrieben, so daß Vorkenntnisse in dieser Sprache geraten scheinen.

Im Springer-Verlag Berlin (nicht Axel) erscheinen Serien von Büchern, die sich mit den Fortschritten und Entwicklungen der Computergrafik beschäftigen. Ein solches, zweibändiges Werk sei hier herausgegriffen: Serie **'EurographicSeminars'**, **'Advances in Computer Graphics'**, herausgegeben von G. Enderle, M. Grave und F. Lillehagen. Eine ganze Reihe von Autoren berichtet hier in ausführlichen Aufsätzen über neue Entwicklungen der Computergrafik. Außerdem enthält der erste Band sehr gute Zusammenfassungen von Grundlagenwissen. Die Beiträge sind in Englisch, allerdings ist auch ein französischer Artikel von kanadischen Autoren darunter. Besonders interessant an diesen Bücherserien von Springer ist, daß sie einen Einblick in aktuelle Themen und Entwicklungen geben, der in Zeitungsartikeln in dieser Ausführlichkeit nicht möglich ist. Leider sind diese Bücher auch sehr teuer.

So, das war's für diese Folge. In einer der nächsten Ausgaben geht es dann weiter. Wer Lust hat: Unbedingt mit Vektoren und Matrizen beschäftigen, es lohnt sich!

(CS)

Info-Kasten 1

Es gibt sehr unterschiedliche grafische Ausgabemedien, zum Beispiel Drucker, Plotter oder Bildschirme. Dabei gibt es zwei Typen von Geräten: Raster- und vektororientierte.

Rasterorientierte Geräte bauen ihr Bild aus einzelnen Punkten auf, vektororientierte Geräte aus Linien. Ein Matrixdrucker muß zum Beispiel eine schräge Linie aus einzelnen Punkten zusammenbauen, was zu Treppchen führt. Ein Plotter dagegen kann direkt eine schräge Linie zeichnen. Rasterorientierte Systeme haben den Vorteil, daß es erheblich einfacher ist, flächige Bilder zu erzeugen.

Auch bei Bildschirmen gibt es vektororientierte Geräte. Diese haben für die Computergrafik aber nicht mehr die Bedeutung, die heute Rasterbildschirme haben. Daher werden wir uns in dieser Serie auf rasterorientierte Ausgabegeräte beschränken. Alle Verfahren, die wir beschreiben, können daher ohne Probleme auf Matrixdruckern oder 'normalen' Computerbildschirmen verwendet werden.

Einkaufsführer

Hier finden Sie Ihren Atari Fachhändler

ANZEIGENSCHLUSS für Heft 10/87 ist der 21.08.87

1000 Berlin



Computare
Keithstr. 18-20 • 1000 Berlin 30
030/21 390 21
186 346 com d

DATAPLAY
Bundesallee 25 · 1000 Berlin 31
Telefon: 030/861 91 61



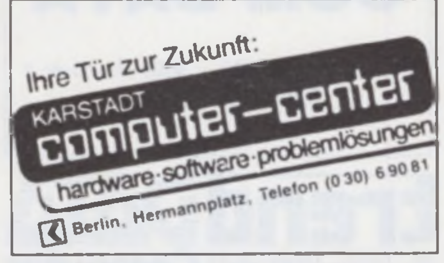


Verkaufsbüro (1.OG) 1 Berlin 12
Knesebeckstr. 78 · Tel. 8827781
Software Hardware Beratung Zubehör Service Literatur



1000 Berlin





Computershop Edith Behrendt
Fürbringerstraße 21 · 1000 Berlin 61
Telefon 0 30/691 76 66

ATARI
... wir machen Spitzentechnologie preiswert.
Vertragshändler
UNION ZEISS
Kurfürstendamm 57 • 1000 Berlin 15
Telefon 32 30 61

2000 Hamburg

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/49 44 00



Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 0 40/641 50 81

2000 Hamburg

Gerhard u. Bernd Waller GbR
Computer & Zubehör-Shop
Kieler Straße 623
2000 Hamburg 54
040/570 60 07 + 570 52 75

Hardware
Software
Beratung
Service



HABA-TECHNIK
Harm Bastian Harms KG
ATARI Systemfachhändler
Münsterstraße 9 · 2000 Hamburg 54
Telefon 040/56 60 1-1

2000 Norderstedt



2120 Lüneburg

Sienknecht
Bürokommunikation
Beratung - Verkauf - Werkstatt
Heiligengeiststr. 20, 2120 Lüneburg
Tel. 04131/46122, Btx 402422
Mo.-Fr. 9⁰⁰-18⁰⁰ und Sa. 9⁰⁰-13⁰⁰

2210 Itzehoe

Der Computerladen
Inhaber Ulrich Bübel Martin Kopplow
Coriansberg 2 · 2210 Itzehoe
Telefon (0 48 21) 33 90/91

2300 Kiel



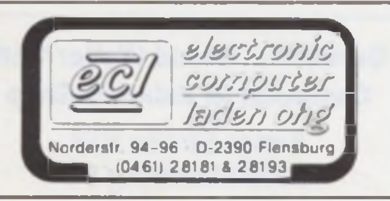
Die Welt der Computer
Dreiecksplatz Nr. 7
2300 Kiel 1 · ☎ 04 31 / 56 70 42

2350 Neumünster



Klosterstraße 2 · 2350 Neumünster
Telefon (0 43 21) 4 39 33

2390 Flensburg

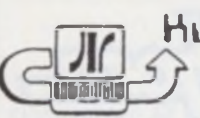


2800 Bremen

PS-DATA
Doventorsteinweg 41
2800 Bremen
Telefon 04 21 - 17 05 77

2850 Bremerhaven

HEIM- UND PERSONALCOMPUTER



Kurt Neumann
Georgstraße 71
2850 Bremerhaven
Tel. 04 71 / 30 21 29

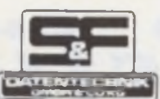
HARDWARE · SOFTWARE · PAPIERWARE

ST-Computer Einkaufsführer
Werbewirksam, aktuell und preiswert.
Sprechen Sie mit uns:
Heim-Verlag
0 61 51 / 5 60 57

2940 Wilhelmshaven

Radio Tiemann
ATARI-Systemfachhändler
Markstr. 52
2940 Wilhelmshaven
Telefon 0 44 21 - 2 61 45

2950 Leer



- HARDWARE-SOFTWARE
- SYSTEM-ENTWICKLUNG
- ORGANISATION
- EDV-SCHULUNG
- EDV-BERATUNG
- SERVICE-WARTUNG

Augustenstraße 3 · 2950 Leer
Telefon 04 91 - 45 89

3000 Hannover



Großer Hillen 6 · 3000 Hannover 71
Telefon (0511) 52 25 79

DATALOGIC COMPUTERSYSTEME
ATARI ST · BERATUNG
COMPUTER · SERVICE
HARDWARE · VERKAUF
SOFTWARE
CALENBERGER STR. 26
3000 HANNOVER 1
TEL. 0511 · 32 64 89

COM DATA
Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11 - 32 67 36

trendDATA Computer
IBM · EPSON · TRIUMPH ADLER
HEWLETT PACKARD · ATARI etc.
trendDATA Computer GmbH
Am Marstall 18-22 · 3000 Hannover 1
Telefon (05 11) 1 66 05 - 0

3040 Soltau

F & T Computervertrieb
Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 0 51 91 / 1 65 22

3100 Celle

Ludwig Haupt jr.
Büro-Einkaufs-Zentrum
Gerhard-Kamm-Straße 2
Ruf 8 30 45, Postfach 140
3100 Celle

3150 Peine

Wieckenberg & Schrage GmbH
Computertechnik
Hard- u. Software
Woltorfer Str. 8, 3150 Peine
Tel. 0 51 71 / 60 52/3 o. 0 51 73 / 79 09

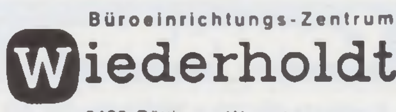
3170 Gifhorn

COMPUTER-HAUS GIFHORN
Braunschweigerstr. 50
3170 Gifhorn
Telefon 0 53 71 - 5 44 98

3300 Braunschweig

COMPUTER STUDIO
BRAUNSCHWEIG
Rebenring 49-50
3300 Braunschweig
Tel. (05 31) 33 32 77 / 78

3400 Göttingen



3400 Göttingen-Weende
Wagenstieg 14 – Tel. 05 51 / 38 57-0

3500 Kassel

Hermann Fischer GmbH
autorisierter ATARI-Fachhändler
Rudolf-Schwander-Str. 5-13
3500 Kassel
Tel. (05 61) 70 00 00

3550 Marburg

L W M
COMPUTER SERVICE
Bahnhofstraße 26b
3550 Marburg/Lahn
✆ 0 64 21 - 6 22 36

4000 Düsseldorf

BERNSHAUS GmbH
Bürotechnik – Bürobedarf
Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81

4000 Düsseldorf

HOCO
EDV ANLAGEN GMBH
Flügelstr. 47
4000 Düsseldorf
Tel. 02 11 - 77 62 70

4050 Mönchengladbach



Hindenburgstr. 249
4050 Mönchengladbach
Tel. 0 21 61 - 1 87 64

4150 Krefeld



- Service-Center
- Hardware
- Software
- Erweiterungen

COP
Computer Service GmbH
Lewerentz Str. 111
4150 Krefeld
Tel. 0 21 51 / 77 30 42

4200 Oberhausen

LaSch
das Buch und Software Haus
Inh. Rainer Langner u. Franz Schnitzler GbR

Nohlstraße 76 · 4200 Oberhausen 1
Telefon 02 08 / 80 90 14

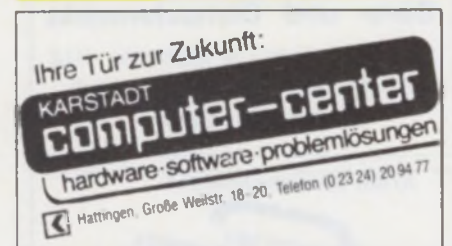
4300 Essen

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Limbecker Platz 4300 Essen 1
Tel.: (02 01) 17 63 99

4320 Hattingen



ST-Einkaufsführer



aktuell
preiswert
werbewirksam

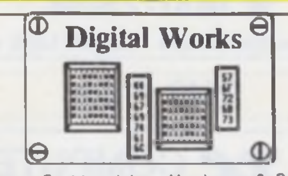


4400 Münster

BASIS
COMPUTER SYSTEME GMBH
Daimlerweg 39 - 4400 Münster
Telefon 02 51 / 71 99 75 - 9

Alle bisher erschienenen Ausgaben ST-Computer gibt's natürlich bei Ihrem Fachhändler

4410 Warendorf



Computer-Fachhandel – Hardware & Software
Jörg Kellert – Helmut Müller GbR
Brünebrede 17 · 4410 Warendorf
Tel. 0 25 81 / 6 11 26

4422 Ahaus

ATARI · Epson · Fujitsu Molecular · NCR · Tandon · Schneider · Star

OCB

OCB-Computershop	OCB-Hard- und Software
Wallstraße 3	Wessumerstraße 49
4422 Ahaus	4422 Ahaus
Tel. 0 25 61 / 50 21	Tel. 0 25 61 / 50 21

4430 Steinfurt

ATARI SCHNEIDER STAR NEC SEIKOSHA PANASONIC EPSON

Computer
Büromaschinen
Service

Telefon 02551/2555

Tecklenburger Str. 27 4430 Steinfurt

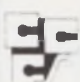
4500 Osnabrück

Heinicke-Electronic
Kommenderiestr. 120 · 4500 Osnabrück
Telefon 05 41 - 8 27 99
Wir liefern Micro-Computer seit 1978

4600 Dortmund

Bürostudio BOLZ
Brauhausstraße 4 · 4600 Dortmund
Telefon 02 31 - 52 77 13-16

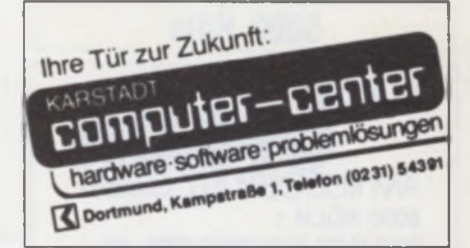
4600 Dortmund



Atari, Genie, Schneider, Tandy, Brother, Star, Memorex, BASF, Verbatim
cc Computer Studio GmbH
Software-Hardware-Beratung
Service-Eilversand

Ihre Ansprechpartner:
v. Schablinski
Jan P. Schneider T. 0231/528184

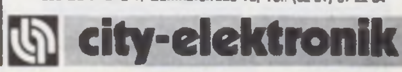
Elisabethstraße 5
4600 Dortmund 1
Tx 822631 cccsd



Elektronik
Computer
Fachliteratur

ATARI-System-Fachhändler

4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84

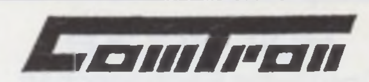


4620 Castrop-Rauxel

R. Schuster Electronic
OBERE MÜNSTERSTR. 33-35 ☎ (0 23 05) 3 77 0 · 4620 CASTROP-RAUXEL

ATARI
System-Fachhändler

4650 Gelsenkirchen-Horst



Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststr. 15 · 4650 Gelsenkirchen-Horst
Tel. 0209/52572

4700 Hamm



4800 Bielefeld

hardware
software
organisation
service

CSF

CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106 – 108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

4800 Bielefeld

Carl-Severing-Str. 190
4800 Bielefeld 14

Telefon: 05 21/45 99-150
Telex : 9 37 340 krab d
Telefax: 05 21/45 99-123

MICROTEC
Software
Hardware
Beratung
Service

5000 Köln

BÜRO MASCHINEN
braun
AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
RUF: 02 21/21 91 71

5010 Bergheim

Computerstudio
HÖLSCHER

EDV-Beratung · Organisation
Programmierung · Home/Personal-Computer
Software · Zubehör · Fachliteratur

Zeppelinstr. 7 · 5010 Bergheim
Telefon 0 22 71 - 6 20 96

5060 Bergisch-Gladbach

Computer Center
Buchholzstraße 1
5060 Bergisch-Gladbach
Telefon 0 22 02 - 3 50 53

5090 Leverkusen

Rolf Rocke
Computer-Fachgeschäft
Auestraße 1
5090 Leverkusen 3
Telefon 0 21 71/26 24

5200 Siegburg

Computer Center
Luisenstraße 26
5200 Siegburg
Telefon 0 22 41/6 68 54

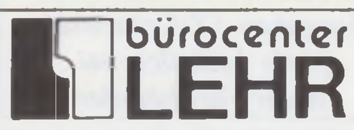
5400 Koblenz

SCHMITT
COMPUTERSYSTEME
Casinostraße 40
5400 Koblenz
✆ 02 61 - 3 65 28

5457 Straßenhaus

DR. AUMANN GMBH
Computer-Systeme
Schulstr. 12
5457 Straßenhaus
Telefon 0 26 34 - 40 81/2

5500 Trier



Güterstr. 82 - 5500 Trier
☎ 06 51 - 2 50 44
Fordern Sie unsere Zubehör-Liste an.

5540 Prüm

ATC COMPUTER
J. M. ZABELL
Ritzstraße 13 · Pf. 10 51
5540 PRÜM
– Tel.: 0 65 51 - 30 39 –

5600 Wuppertal

Jung am Wall
Wall 31—33
5600 Wuppertal 1
Telefon 02 02/45 03 30

MEGABYTE
Computer Vertriebs GmbH

Friedrich-Engels-Allee 162
5600 Wuppertal 2 (Barmen)
Telefon (02 02) 8 19 17

5630 Remscheid

C O M SOFT
Scheiderstr. 12 · 5630 Remscheid
Telefon (0 21 91) 2 10 33 - 34

5800 Hagen



Vertragshändler **Axel Böckem**
Computer + Textsysteme

Eilper Str. 60 (Eilpezentrum) · 5800 Hagen
Tel. 0 23 31/7 34 90

5900 Siegen

HeesComputer
Vertriebs GmbH
Hardware Software Schulung
Siegen · Weidenauer Str. 72 · ☎ 02 71/7 34 95

6000 Frankfurt

Müller & Nemecek
Kaiserstraße 44
6000 Frankfurt/M.
Tel. 0 69 - 23 25 44

GES-COMPUTER
GESELLSCHAFT FÜR EDV UND SOFTWARE mbH

Filiale Frankfurt
Hartmann-Ibach-Str. 63
6000 Frankfurt 60
Tel.: (0 69) 46 20 41

Filiale Hanau
Steinheimer Str. 22
6450 Hanau
Tel.: (0 61 81) 2 48 26

WAIZENEGGER
Büroeinrichtungen

Kaiserstraße 41
6000 Frankfurt/M.
☎ 0 69/2 73 06 - 0

ATARI
... wir machen Spitzentechnologie preiswert.
Vertragshändler
jetzt bei uns.
bco
Büro-Computer + Organisations GmbH
Oederweg 7—9
6000 Frankfurt/M. 1
☎ (0 69) 55 04 56-57
Hardware ✱ Software ✱ Beratung ✱ Service

SCHMITT
COMPUTERSYSTEME
Hanauer Landstr. 417
6000 Frankfurt/M. 1
Tel: 0 69/41 02 71
FAX: 0 69/42 57 71

6100 Darmstadt

Heim
Büro- und Computermarkt
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
✆ 0 61 51/5 60 57

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Elisabethenstr. 15 · 6100 Darmstadt
Luisencenter · Tel. 0 61 51 - 10 94 20

6100 Darmstadt

SCHMITT COMPUTERSYSTEME
Mühlstraße 76
6100 Darmstadt
✆ 0 61 51 - 2 45 74

6200 Wiesbaden

SCHMITT COMPUTERSYSTEME
Rheinstraße 41
6200 Wiesbaden
✆ 0 61 21 - 30 73 30

6240 Königstein

KFC COMPUTERSYSTEME
Wiesenstraße 18
6240 Königstein
Tel. 0 61 74 - 30 33
Mail-Box 0 61 74 - 53 55

6300 Gießen



Schneider ATARI Commodore
BAUMS
BÜRO · ORGANISATION
Bahnhofstr. 26 · 6300 Gießen
Telefon: 06 41 / 7 10 96



Thomas Heß
Asterweg 10 · 6300 Gießen
Telefon 06 41 / 3 91 53

6330 Wetzlar



Fachmarkt für Computer u. Unterhaltungs-electronic in Wetzlar.
Einkaufszentrum Bahnhofstraße. Tel. (0 64 41) 4 85 66

6400 Fulda

Schneider ATARI Commodore
WEINRICH
BÜRO · ORGANISATION
Ronsbachstraße 32 · 6400 Fulda
Telefon: 06 61 / 4 92-0

6457 Maintal

Landolt-Computer
Beratung · Service · Verkauf · Leasing
Wingertstr. 112
6457 Maintal/Dörnigheim
Telefon 0 61 81 - 4 52 93

6500 Mainz

:ELPHOTEC
Computer Systeme
Ihr Atari Systemhändler mit eigenem Service-Center
Walpodenstraße 10
6500 Mainz
Telefon 0 61 31 - 23 19 47

SCHMITT COMPUTERSYSTEME
Karmeliterplatz 4
6500 Mainz
✆ 0 61 31 - 23 42 23

6520 Worms

ORION
Computersysteme GmbH
Friedrichstraße 22
6520 WORMS
Tel. 0 62 41 / 67 57 - 67 58

6700 Ludwigshafen

MKV Computermarkt
Bismarck-Zentrum
6700 Ludwigshafen
Telefon 06 21 - 52 55 96

6720 Speyer

MKV Computermarkt
Gilgenstraße 4
6720 Speyer
Telefon 0 62 32 - 7 72 16

6750 Kaiserslautern

C.O.S.
COMPUTER ORG. GmbH
Karl-Marx-Straße 8
6750 Kaiserslautern
Telefon (06 31) 6 50 61 - 62



6800 Mannheim



Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912

Computer-Center
am Hauptbahnhof GmbH
L 14, 16-17
6800 Mannheim 1
Tel. (06 21) 2 09 83 / 84

6900 Heidelberg

JACOM COMPUTERWELT
Hardware · Software
Schulung · Service
Mönchhofstraße 3 · 6900 Heidelberg
Telefon 0 62 21 / 41 05 14 - 550

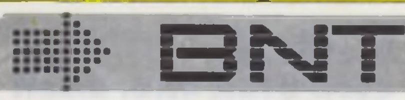


Bahnhofstr. 1 / Tel. 0 62 21 / 2 71 32

7000 Stuttgart

SCHMITT COMPUTERSYSTEME
Tübingerstr. 18
7000 Stuttgart

7000 Stuttgart



COMPUTERFACHHANDEL

Ihr starker Partner in Stuttgart

BNT Computerfachhandel GmbH Marktstr. 48 1. Stock
7000 Stuttgart-Bad Cannstatt in der Fussgängerzone
direkt beim Rathaus Telefon (0711) 55 83 83

7022 L-Echterdingen

Autorisierter ATARI-System-Fachhändler
ATARI ST

Matrai computer

Matrai Computer GmbH
Bernhauser Str 8
7022 L.-Echterdingen
☎ (07 11) 79 70 49

7030 Böblingen

Verkauf – Service – Software



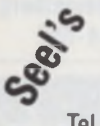
Norbert Hlawinka
Sindelfinger Allee 1
7030 Böblingen
Tel. 0 70 31 / 22 60 15

COMPUTER CENTER

7100 Heilbronn

Unser Wissen ist Ihr Vorteil

Walliser & Co.
Mönchseestraße 99
7100 Heilbronn
Telefon 07131/60048



Computer-Welt
Am Wollhaus 6
7100 Heilbronn
Tel. 0 71 31 - 6 84 01 - 02

7150 Backnang



7410 Reutlingen

Computer-Shop
Werner Brock
autorisierter Fachhändler f. ATARI, Schneider, Oki

Federnseestr. 17 · 7410 Reutlingen
Telefon: 0 71 21 / 3 42 87

7450 Hechingen

SRE
Gesellschaft für Datenverarbeitung mbH

Computer · Drucker
Zubehör · Fachliteratur

Schloßplatz 3 · 7450 Hechingen
Telefon 0 74 71 / 1 45 07

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis
HEIM + PC-COMPUTERMARKT
HARDWARE · SOFTWARE · LITERATUR
SCHEURER
ATARI COMMODORE CUMANA DATA-BECKER
MULTITECH RITEMAN SCHNEIDER THOMSON

7475 Meßstetten 1 · Hauptstraße 10 · 0 74 31 / 6 12 80

7480 Sigmaringen



7500 Karlsruhe

papierhaus erhardt

Am Ludwigsplatz · 7500 Karlsruhe
Tel. 07 21 - 2 39 25

MKV GMBH

Kriegsstraße 77
7500 Karlsruhe
Telefon 07 21 - 8 46 13

7530 Pforzheim

DM Computer GmbH
Hard- & Software
Durlacherstr. 39
Tel. 0 72 31 - 1 39 39
7530 Pforzheim
Telex 783 248

7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk

In der Jeuch 3
7600 Offenburg
Telefon 07 81 / 5 79 74

7640 Kehl/Rhein



Computer · Software · Marketing
eigener Service · eigene Software

Badstrasse 12
Tel. 0 78 07 / 8 22
Telex: 752 913
7607 NEURIED 2

Filiale:
Hauptstrasse 44
Tel. 0 78 51 / 18 22
7640 KEHL/RHEIN

ELEKTRO-MÜNTZER GmbH

7700 Singen

U. MEIER
Computersysteme
7700 Singen-Htwl.
Am Posthalterswäldle 8
Telefon 0 77 31 - 4 42 11

7730 VS-Schwenningen

BUS BRAUCH & SAUTER COMPUTER TECHNIK
Villinger Straße 85
7730 VS-Schwenningen
Telefon 0 77 20 / 3 80 71 - 72

7750 Konstanz

ATARI ★ PC's ★ SCHNEIDER

computer - fachgeschäft
rösler
Rheingutstr. 1 · ☎ 0 75 31 - 2 18 32

7800 Freiburg

CDS
EDV-Service GmbH
Windausstraße 2
7800 Freiburg
Tel. 07 61 - 8 10 47

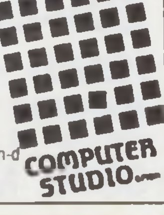
7890 Waldshut-Tiengen

hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 0 77 51 / 30 94

7900 Ulm

HARD AND SOFT COMPUTER GMBH
Ulms großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Herrenkellergasse 16 · 7900 Ulm/Donau
Telefon 07 31 / 6 26 99

7900 Ulm



EDV-Systeme
Software-erstellung
Schulung

Systemhaus:
Frauenstr. 28
7900 Ulm/Donau
Tel. 07 31/2 80 76
Telex 712 973 csulm-d

7950 Biberach

HARD AND SOFT COMPUTER GMBH

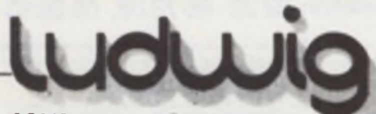
Biberachs großes Fachgeschäft für BTX, Heim- u. Personalcomputer
Schulstraße 6 · Bei der VHF
7950 Biberach · Tel. 0 73 51/1 22 21

7980 Ravensburg

GRAHLE
Expert Grahle Computer
Eisenbahnstr. 33
7980 Ravensburg
Tel.: 0751/15955

Vertragshändler für ATARI, Schneider und Star

8000 München



COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTADTER STR. 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3113066 · TELETEX 898341

SCHMITT COMPUTERSYSTEME

Arnulfstraße
8000 München



ATARI COMPAQ
Das Computer-Fachgeschäft im HERTIE · Hauptbahnhof
Bahnhofsplatz 7, 8000 München 2
Telefon 089/595277
Ein Unternehmen der -Gruppe

schulz computer

Schillerstraße 22
8000 München 2
Telefon (0 89) 59 73 39
Beratung · Verkauf · Kundendienst

8032 Gräfeling



COMPUTER SYSTEME SCHULUNG
Am Haag 5
8032 Gräfelfing
Tel. 089-854 54 64, 85 10 43

8100 Garmisch-Partenk.

Uwe Langheinrich Elektronik Center

Hindenburgerstr. 45
8100 Garmisch-Partenkirchen
Tel. 0 88 21-7 15 55
Bitte Gratisliste anfordern

8120 Weilheim

Klement

COMPUTER FACHHÄNDLER

Elektro-, Radio- und Fernseh-Center
Beleuchtungskörper · Schallplatten
Meisterbetrieb

8120 Weilheim · Admiral-Hipper-Straße 1
Geschäft ✆ 45 00 · Kundendienst ✆ 44 00
Computer-Studio ✆ 6 21 67
Interfunk-Fachgeschäft

8150 Holzkirchen

ATARI
Besuchen Sie uns!
Fordern Sie unseren Software-Katalog (520ST) an!

MÜNZENLOHER GMBH
Tölzer Straße 5
D-8150 Holzkirchen
Telefon: (0 80 24) 18 14

8170 Bad Tölz

Uwe Langheinrich Elektronik Center

Wachterstr. 3
8170 Bad Tölz
Tel. 0 80 41-4 15 65
Bitte Gratisliste anfordern

8330 Eggenfelden



Computer-Centrum
R. Lanfermann
Schellenbruckstraße 6
8330 Eggenfelden
Telefon 0 87 21/65 73

Altöttinger Straße 2
8265 Neuötting
Telefon 0 86 71/7 16 10

Innstraße 4
8341 Simbach
Telefon 0 85 71/44 10

8400 Regensburg

C-SOFT GMBH
Programmentwicklung & Hardware
Holzfällerstraße 4
8400 Regensburg
Telefon 09 41/8 39 86

Zimmermann elektroland

8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41/9 50 85

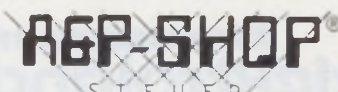
8390 Passau
Kohlbruck 2a
☎ 08 51/5 20 07

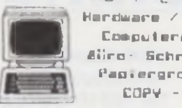
8423 Abensberg

COMPUTERVERSAND WITTICH
Tulpenstr. 16 · 8423 Abensberg
☎ 0 94 43/453



8490 Cham





Hardware / Software
Computerzubehör
Büro- Schreibwaren
Papiergroßhandel
COPY - SHOP

Auf der Schanze 4 · 8490 Cham/Opf. · Telefon (0 99 71) 97 23

8500 Nürnberg

EINE IDEE ANDERS
KARSTADT NÜRNBERG AN DER LORENZKIRCHE



1. KLASSE EINKAUFEN IM WELTSTADTHAUS

SCHMITT COMPUTERSYSTEME

Innere Laufer Gasse 29
8500 Nürnberg
✆ 09 11-20 97 17

ATARI

Der ATARI - Systemfachhändler in Nürnberg
Softwareberatung und Entwicklung
Computer · Drucker · Zubehör · Fachliteratur



HIB Computer GmbH
Äußere Bayreuther Str. 72
8500 Nürnberg
Tel.:0911/515939

CH-3415 Hasle Rüegsau

COMPU-TRADE
Ihr ATARI Spezialist
Emmenstr. 16
CH-3415 Hasle-Rüegsau ·
✆ 0 34 / 61 45 93 auch abends bis 21.00 h
HARD- u. SOFTWARE · BERATUNG · EILVERSAND

CH-4625 Oberbuchsiten

STECTRONIC M. Steck
Electronic-Computer-Shop
Hauptstr. 104/137
CH-4625 OBERBUCHSITEN
Tel. 0 62 / 63 17 27 + 63 10 27

CH-5430 Wettingen



Senn Computer AG
Zentralstrasse 93
CH-5430 Wettingen
Tel. 056 / 27 16 60
Telex 814 193 seco

CH-8006 Zürich

ADAG Computershop
Universitätsstr. 25 · 8006 Zürich
Tel. 01 / 47 35 54
ATARI & WANG & EPSON

Computer-Center P. Fisch
Stampfenbachplatz 4
8006 ZÜRICH
✆ 01 / 363 67 67

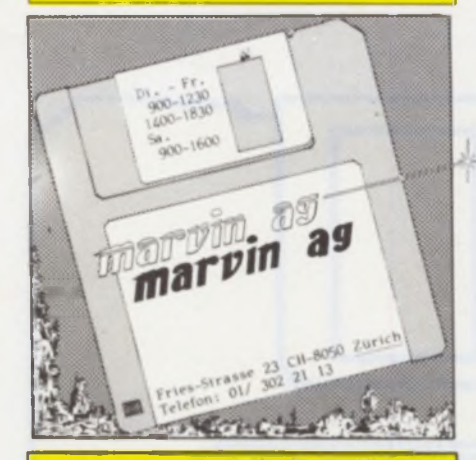
CH-8021 Zürich



Senn Computer AG
Langstrasse 31
Postfach
CH-8021 Zürich
Tel. 01 / 241 73 73
Telex 814 193 seco

VILAN
Das Warenhaus der neuen Ideen
Ihr Computer-Fachhandel an der
Bahnhofstrasse 75 · Zürich

CH-8050 Zürich



CH-8200 Schaffhausen

ZIMELEC
CAR-AUDIO
COMPUTER + ELECTRONIC
Bachstraße 28 · 8200 Schaffhausen
Tel.: 053/55224

Montag—Freitag	9.00—12.00 13.30—18.30
Samstag	9.00—16.00

CH-9000 St. Gallen

VIDEO - COMPUTER - CENTER
GÄCHTER AG
Webergasse 22
9000 St. Gallen
Telefon 071 / 22 60 05



CH-9400 Rorschach



Hier könnte
Ihre Anzeige
erscheinen.
Anruf genügt:
Heim-Verlag
✆ 06151 - 56057

Luxemburg



Der Einkaufsführer in ST-Computer



aktuell
preiswert
werbewirksam



Anzeigenschluß:

Heft 10/87: 21.08.87
Heft 11/87: 23.09.87
Heft 12/87: 20.10.87



Bevor der Floppykurs so richtig in die Vollen geht, drehen wir noch eine theoretische Ehrenrunde. Es geht um das Drumherum des Floppycontrollers, dessen Programmierung sich bis jetzt kaum jemand zutraut. Aber vielleicht wird dank dieser Serie bald der Kopierschutz und die Datenpackerei auf der Diskette zum Volkssport – wenn Sie ein bißchen durchhalten.

Zur Unterstützung Ihrer eigenen Experimente gibt es eine Erweiterung für den Diskmonitor MINIMON, den ich in der letzten ST vorgestellt habe. Sie wird aus Platzgründen leider erst im nächsten Heft abgedruckt. Die Schwierigkeit dabei ist, daß Floppyprogrammierung nur in Assembler denkbar ist – aber keine Angst, Sie müssen sich nicht monatelang mit **moveq, lea** und Konsorten herumschlagen. Hauptsache ist, daß Sie die Assemblerroutinen, die in das BASIC-Programm eingebunden werden, anwenden können. Und dazu muß ich wohl oder übel auch ein wenig trockene Theorie erklären.

"Kontrola ... was'n das?"

Das mögen sich einige Leser fragen, die bis jetzt in diesem Kurs noch nicht dabei waren oder die nicht so recht aufgepaßt haben. Der **Floppycontroller** ist ein Chip, der nur für den Datenaustausch mit dem Laufwerk zuständig ist. Er entlastet den Prozessor, indem er ihm die lästige Arbeit abnimmt, die Diskettendaten zu ver- und entschlüsseln sowie sie zu lesen und zu schreiben.

Zunächst einige Worte über die Aufzeichnung. Daß eine Diskette in Spuren und Sektoren aufgeteilt ist, wissen Sie. Sehen wir uns so eine Spur an (siehe Bild 1).

Sie sehen, daß Sie nichts sehen. Das ist auch ganz in Ordnung so, denn schließlich haben wir der Spur noch keine Informationen für ihr künftiges Leben mitgegeben. Nun könnten Sie sich vorstellen, die Bits würden einfach nacheinander auf die Spur gekleistert: Magnetisierter Fleck für „1“, nicht magnetisierter für „0“. Nicht schlecht für den Anfang, aber unbrauchbar. Denn so eine Spur ist rund, und jetzt erzählen Sie mir mal, wo ein Kreis beginnt. Eine Markierung anbringen, nach jeder Umdrehung des Laufwerksmotors einen Indeximpuls erzeugen? Gute Idee, wird auch tatsächlich praktiziert, nur reicht das noch nicht. Ein Laufwerk kann mit vernünftigem technischen Aufwand nicht dazu gebracht werden, immer gleich schnell zu drehen.

TAKTLOSIGKEITEN

Irgendwie muß man also den Lesevorgang mit der Geschwindigkeit des Laufwerks synchronisieren. Ein Verfahren dazu heißt MFM, es wird beim ST und bei den meisten anderen Rechnern angewendet und codiert den Taktimpuls des Controllers mit in die Datenbits ein. Schreiben wir einmal einen solchen Track (siehe Bild 2).

Jetzt können wir also einwandfrei Bits lesen, ohne taktlos zu werden. Geschafft? Denkste. Wie erkenne ich denn jetzt, wo ein bestimmtes Byte anfängt? Natürlich, Indeximpuls abwarten, Bits lesen und dabei immer bis 8 Bits (= 1 Byte) zählen, bis man beim gewünschten Byte ankommt... Nein, das ist es immer noch nicht. Man will ja nicht immer den ganzen Track lesen, sondern vielleicht nur einen Sektor, und außerdem könnte es ja sein,

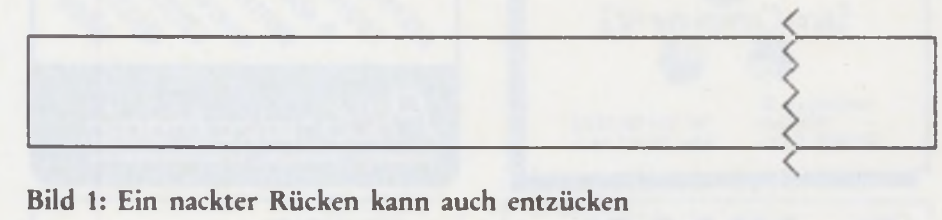


Bild 1: Ein nackter Rücken kann auch entzücken



Bild 2: Bit für Bit – Schreiben in Salamitaktik

daß die Synchronisation mit dem Indeximpuls nicht ganz so glatt läuft; dann fängt der Controller beim falschen (zum Beispiel beim zweiten) Bit zu lesen und zu zählen an. Ergebnis: Lesefehler, Bombenanschläge, intergalaktische Kollisionen.

In der Praxis schreibt man spezielle Bytes auf die Spur, die Synchronisationsbytes. Diese Bytes erkennt der Controller daran, daß sie ohne Taktinformationen geschrieben werden. Meistens schreibt man mehrere Sync-Bytes hintereinander, und der Controller verschlingt solange Bits, bis er mal einen gültigen Wert für ein Syncbytes ($A1) erkannt hat. Dann weiß er: Hoppla, das nächste Bit ist der Anfang eines Bytes. Hier also unser Track, wie er wirklich aussieht (das heißt, immer noch etwas verbogen und nicht so schön rund wie in Wirklichkeit) (siehe Bild 3).

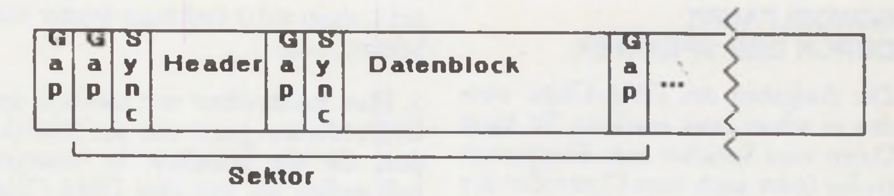


Bild 3: Die Wahrheit kommt ans Licht

Die Markierungen der einzelnen Bits habe ich der Übersicht wegen weggelassen. „Gap" bezeichnet einen Abschnitt von Lückenbytes, die der Controller braucht, um sich zwischen den Daten zu erholen und klar Schiff zu machen. Dabei ist er allerdings fixer, als Atari und sogar sein Hersteller ihm zutrauen; darauf beruht meine Idee, die Lückenbytes soweit zu kürzen, bis ein elfter Sektor auf die Spur paßt (HYPERFORMAT).

Nach den Syncbytes folgen die sogenannten Adreßmarken, mit denen bestimmte Datenstrukturen angekündigt werden (Datenheader, also Vorspann oder Daten). Auch sie werden ohne Taktimpulse geschrieben. Nach den Daten- bzw. Vorspannblöcken folgt jeweils eine 2-Byte-Checksumme, mit der man die Gültigkeit der Informationen nachprüfen, aber auch jede Menge Unsinn treiben kann.

HACKORDNUNGEN ODER DER NAME DER DOSE

Die ganze Lese-/Schreibarbeit und die Codierungsvorgänge werden dem Abt des Klosters Atari ST, Prozessorus Maximus, von seinem Schreiberling Floppius Controllus abgenommen. Nun ist aber Prozessorus auch noch zu bequem, um Controllus die Daten selbst zu bringen oder sie abzuholen: Er setzt einen Boten ein, D.M.A. Controllus genannt, der speziell für Datenschaufeleien dieser Art angeheuert wurde. Ernst beiseite: Der DMA-Controller (Direct Memory Access) bekommt beim Schreiben vom Prozessor die Instruktion, was der Floppycontroller tun soll. Hier dient er als Relaisstation; außerdem bekommt er eine Speicheradresse, wo die Informationen stehen, die er dem Controller Häppchen für Häppchen selbständig übermitteln soll. Der Prozessor zieht sich solange würdevoll zurück, um das Ende der Übertragung abzuwarten, das er entweder im Erreichen einer bestimmten DMA-Adresse oder an einer Unterbrechungsanforderung des Controllers merkt. Das Lesen funktioniert analog.

Warum so ein Aufwand? Während der DMA-Chip sich um die Speicherschaufelei kümmert, kann der Prozessor sich seiner Freizeit oder wichtigeren Aktivitäten widmen. Vor allem für Multitasking-Betriebssysteme ist das überlebenswichtig. Nur leider nützt TOS diese Möglichkeiten nicht aus und verbrät die kostbare Prozessorzeit in Warteschleifen, solange gelesen und geschrieben wird. In RTOS dagegen kann man sogar während des Formatierens noch andere Dinge erledigen! Ein Schema soll die folgenden, recht kniffligen Ausführungen erhellen: (siehe Bild 4).

Es zeigt alle Chips, die bei der Ein-/Ausgabe auf Diskette zusammenspielen. Nicht nur Prozessor, DMA-Controller und Floppycontroller sind – wie man sieht – am Lesen/Schreiben beteiligt. Die Unterbrechungsanforderung des Floppycontrollers, mit der er vehement auf das Ende seiner Arbeit hinweist, gelangt zum MFP68901, einem Baustein im ST, der unter den Chips des ST nur „der Störenfried" heißt, weil er ständig andere bei der Arbeit unterbricht: Er verwaltet alle Interruptanforderungen der Peripherie.

Dazu kommt noch, daß der Controller eigentlich nur ein Laufwerk ansprechen kann – und sogar nur eine Seite! Aber hilfsbereit, wie er ist, springt da der Soundchip ein und leiht dem Controller drei seiner Portbits und schaltet mit ihnen je nach Bedarf zwischen den Seiten und den Laufwerken (maximal zwei) um.

Wie diese Chips zusammenwirken und wie man sie dazu bringt, sich gegenseitig zu unterstützen, wird uns nun beschäftigen. Die Dreh- und Angelpunkte beim Diskzugriff sind Floppycontroller (auch FDC für Floppy Disk Controller) und DMA-Controller.

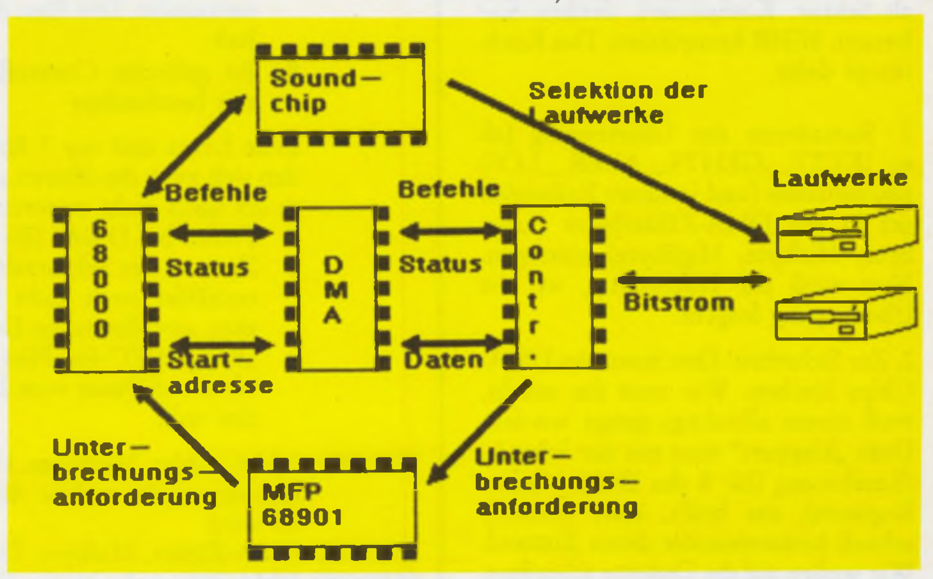


Bild 4: Die Maikäfer kommen!

GONDELFAHRT DURCH DEN SPEICHER

Die Aufgaben des DMA-Chips wurden ja schon kurz erwähnt: Er hievt Daten vom Speicher zum Floppycontroller (oder auch zum Controller der Festplatte) und von dort aus wieder in den Speicher, ohne daß sich der Prozessor darum kümmern muß. Der Prozessor gibt dem DMA-Chip seine Anweisungen, was er gerne da und dort hätte, und beschäftigt sich dann lieber damit, Invaders oder StripPoker zu spielen.

Die Anweisungen des Prozessors landen in ein paar Adressen, die ich jetzt in ihre Bits zerlegen werde. Auch die Ausgabe des DMA-Chips (Statusmeldungen und ähnliches) läuft über diese Adressen (siehe Tabelle 1).

WIR KÖCHELN UNS EIN DATENBRÄU

Wie geht man mit diesem Wust nun um? Einzelne Register des Floppycontrollers anzusprechen, ist ja noch relativ einfach: Man schreibt zum Beispiel den Wert $80 in das Modusregister des DMA-Chips. Damit hat man das Statusregister des Floppycontrollers ausgewählt, das man dann im Zugriffsregister $FF8604 auslesen kann. Anderes Beispiel: $184 ins Modusregister, und schon kann man in $FF8604 das Sektorregister beschreiben (halten Sie sich immer die Tabelle 1 vor Augen!). Und so weiter. Nehmen wir an, wir wollten jetzt den Speicherbereich von JETZT_GEHTS_ABER_LOS bis DA_HÖRTS_ABER_AUF auf die Diskette schreiben, beispielsweise als Sektor. Kompliziert, denken Sie? Irrtum. SEHR kompliziert. Das Kochrezept dafür.

1. Startadresse der Übertragung (also JETZT_GEHTS_ABER_LOS) nacheinander (und in dieser Reihenfolge) in die DMA-Zählerbytes (Lowbyte, Midbyte, Highbyte) eintragen. Nun weiß der DMA-Chip, wo die Übertragung losgeht.

2. Zur Sicherheit: Den Status des DMA-Chips löschen. Wie man das macht, muß einem allerdings gesagt werden: Dazu „klappert" man mit der Schreib-/Leseleitung (Bit 8 des DMA-Modus-Registers), das heißt, man wechselt schnell hintereinander deren Zustand. Wir wollen auf die Diskette schreiben, also legen wir die Leseleitung zuerst auf 1, dann auf 0 und dann wieder mit Schritt 3 auf 1.

3. Hier beschreiben wir nämlich das DMA-Modusregister mit der Instruktion, die wir brauchen. In unserem Fall wollen wir mit dem DMA-Chip Daten übertragen. Wo er mit der Übertragung anfangen soll, weiß er bereits. Aber wann hört er auf? Das müssen wir ihm noch in sein goldenes Buch, den Sektorzähler schreiben. In den Sektorzähler schreibt man die Anzahl der zu übertragenden Sektoren (ein Sektor besteht auch hier aus 512 Bytes), der DMA-Chip zählt dieses Register dann selbständig während der Übertragung herunter. Indem Sie $190 ins DMA-Modus-Register schreiben, wählen Sie den Sektorzähler des DMA-Chips zum Beschreiben aus.

4. Diesem Sektorzähler übergeben wir in $FF8604 nun die Maximalzahl von

Adresse | Bedeutung — **Tabelle 1**

$FF8604 REGISTERZUGRIFF
Nur die unteren 8 Bits sind belegt. Aus dieser Adresse kann man den Inhalt von DMA-Chip- und Floppycontroller-Registern lesen oder man kann sie durch diese Adresse verändern. Welches Register ausgewählt ist, hängt von der folgenden Adresse ab.

$FF8606 DMA-Modus, DMA-Status
Für den Schreibzugriff gilt folgende Belegung:

Bit	Wirkung
0	keine erkennbare Bedeutung, rätselhaft
1,2	Wenn Bit 4 gelöscht ist, kann man mit diesen Bits die Register des Floppycontrollers auswählen, auf die man dann in $FF8604 zugreifen kann. Dabei spielt auch der Zustand des Schreib-/Lesebits (Bit 8) eine Rolle:
3	Datenaustausch mit dem Floppycontroller (Bit gelöscht) oder mit der Harddisk (Bit gesetzt)
4	Ist das Bit gelöscht, kann man in $FF8604 die Register des FDC manipulieren, ansonsten den „Sektorzähler" des DMA-Chips (siehe unten).
5	Mysteriös
6	Bit gelöscht: DMA an, Bit gesetzt: DMA macht Ferien
7	dasselbe wie Bit 3, nur in grün: die Bitbedeutung ist gerade vertauscht. Der Sinn dieses Doppel-Moppels ist mir schleierhaft.
8	Bit gelöscht: Controllerregister lesen, sonst: Controllerregister beschreiben

Zu Bit 1,2:

Bit 1	Bit 2	angesprochenes FDC-Register
0	0	beim Lesen Statusregister, sonst Kommandoregister
1	0	Spurregister
0	1	Sektorregister
1	1	Datenregister

Beim Lesen sind nur 3 Bits erkennbar belegt, die restlichen melden sich zwar des öfteren als gesetzt, was sie aber bedeuten, konnte ich noch nicht eruieren:

Bit	Bedeutung
0	Fehler bei DMA? (Bit gesetzt bedeutet ja)
1	Zustand des Sektorzählers (Bit gesetzt bedeutet, daß der Sektorzähler noch nicht bei 0 angekommen ist; dadurch kann man unvollständige Datenübertragungen erkennen)
2	„REQUEST"-Bit. Hier telefoniert der Floppycontroller durch, ob er auf Daten vom DMA-Chip wartet oder welche loswerden will.

$FF8609 DMA-Zähler, Highbyte. Hier liegen die obersten 8 Bit eines Zählers, der angibt, welche Adresse vom DMA-Chip gerade bearbeitet wird.

$FF860B DMA-Zähler, Midbyte. Die mittleren 8 Bit.

$FF860D DMA-Zähler, Lowbyte. Die unteren 8 Bit.

Sektoren, die der DMA-Chip übertragen soll. Bei uns wäre das 1 + (DA__HÖRTS__ABER__AUF – JETZT__GEHTS__ABER__LOS) / 512, die 1 wird addiert, um auch die paar überzähligen Bytes noch loszuwerden, falls die Divison nicht glücklich und ganzzahlig ausgeht (und wo gibt es schon noch ein happy end?).

5. Auf ähnliche Weise sagen wir dem Floppycontroller, welchen Sektor er auf der Diskette beschreiben soll. $184 ins DMA-Modusregister, und das Sektorregister des FDC liegt uns in $FF8604 fertig zum heftigen Beschreiben zu Füßen. Was wir dann auch tun.

6. Und jetzt wählt man das Kommandoregister des FDC aus ($180 ins DMA-Modus-Register) und übergibt ihm in $FF8604 einen Read-Sector-Befehl.

Natürlich war's das noch lange nicht. Jetzt müßte man noch das Ende der Übertragung abwarten. Das macht man entweder, indem man die Unterbrechungsanforderung des Floppycontrollers abwartet, mit der er Fehler oder das Ende seiner Mühsal anzeigt, oder indem man den DMA-Zähler beäugt, abwartet, bis die Übertragung bei einer bestimmten Adresse angelangt ist (in unserem Fall bei DA__HÖRTS__ABER__AUF) und dann den Controller brutal abwürgt (dazu gibt es einen speziellen Strangulationsbefehl). Und wer ganz sauber arbeitet, sollte sich danach noch den Status von DMA-Chip und Floppycontroller anschauen, um eventuelle Fehler abzufangen...

Damit Sie sich schnell zurechtfinden und nicht immer in die einzelnen Bits gehen müssen, habe ich in einer Tabelle die Werte zusammengefaßt, die man in die DMA-Modus-Adresse schreiben muß, um bestimmte Register auszuwählen. Hier die Tabelle für das Lesen von Registern:

Wert	angesprochenes Register
$80	Statusregister des Floppycontrollers
$82	Spurregister des FDC (Floppycontroller)
$84	Sektorregister des FDC
$86	Datenregister des FDC
$90	Sektorzähler des DMA-Chips

Und die Tabelle für den Schreibzugriff auf Register:

Wert	Register
$180	Kommandoregister des FDC
$182	Spurregister des FDC
$184	Sektorregister des FDC
$186	Datenregister des FDC
$190	Sektorzähler des DMA-Chips

Eine kleine humoristische Einlage des DMA-Chips sollte noch Erwähnung finden: Liest man weniger als 16 Bytes ein, tut sich gar nichts. Woran liegt's? Der DMA-Chip puffert intern 16 Bytes, bevor er sie in den Speicher schreibt. Wenn wir also 167 Bytes zu lesen wünschen, liefert der DMA-Chip 160 (zehnmal 16) und behält 7. Dieser Puffer wird übrigens gelöscht, wenn Sie den Status per Schreib-/Leseleitung-Klappern löschen...

Beim Schreiben von Datenmengen (also nicht von Registern) benimmt sich der DMA-Chip noch eigenwilliger. Man ist nicht etwa dann mit dem Schreiben eines Sektors fertig, wenn die aktuelle DMA-Adresse 512 Bytes höher ist als die Startadresse, sondern erst bei 512+32 Bytes! Zum Glück spinnt nur der DMA-Zähler und nicht die Datenübertragung selbst, so daß man diesen Fehler leicht umgehen kann: Man wartet einfach ein bißchen länger (eben bis laut Zähler 512+32 Bytes übertragen sind).

Wenn Sie das alles auf Anhieb verstanden haben, sind Sie wahrscheinlich an der Entwicklung des DMA-Chips beteiligt gewesen. Schauen Sie sich mal im Listing 1 (LOCKSLEY.S) die Routine **wrsec** an, mit der man Sektoren schreibt. Dort werden Sie wiederfinden, was ich Ihnen erklärt habe. Noch eine Anmerkung: Bilden Sie sich nicht ein, Sie könnten diese direkte Programmierung des DMA-Chips effektiv und effizient in einer Hochsprache erledigen – wenn es eine Hochburg der Assemblerprogarmmierung gibt, dann liegt sie in der Floppyprogrammierung, wo es zeitlich ziemlich knapp wird.

Aber geben Sie jetzt nicht auf, weil ihre einzigen Fremdsprachen Englisch und BASIC sind. Für Sie stelle ich im nächsten Teil des Floppykurses eine Erweiterung für den MINIMON vor, mit dem Sie all diese Dinge von BASIC aus erledigen können. Bevor ich dazu komme, bringen wir noch schnell die Funktionen von Soundchip und MFP beim Diskzugriff hinter uns.

LUSTVOLLES STÖHNEN NACH FEIERABEND

Wenn der Controller fix und fertig mit der Welt ist, also sein Kommando abgearbeitet hat, meldet er das, indem er eine Unterbrechungsanforderung auslöst. Die meldet er dem MFP (Sie wissen schon, der ungemütliche Zeitgenosse von vorhin). Wie fragt man das ab? Ganz einfach:

```
btst #5,mfp
beq fertig
```

Dieser Assemblerbrocken testet Bit 5 des I/O-Port im MFP (Adresse $FFFA01). Wenn das erschöpfte 'Fertig'-Keuchen des Controllers durch die Chiplandschaft hallt, meldet das der MFP, indem er dieses Bit auf 0 setzt. Wie das im Programm aussieht, sehen Sie in der Routine WARTEN__AUF__GODOT von LOCKSLEY.S (siehe Listing 1).

STABILE SEITENLAGE IM LAUFWERK

Beim Soundchip sind im Grunde nur zwei Adressen interessant. Die eine heißt schlicht SND (zumindest in meinem Select-Programm), liegt bei $FF8800 und ist das Auswahlregister des Chips. Das heißt, man schreibt die Nummer des gewünschten Soundchipregisters in diese Adresse hinein und kann dann entweder deren Inhalt aus SND lesen oder das angewählte Register verändern, indem man in $FF8802 (SNDWRT für „Soundchip-Write-Eingang") einen Wert schreibt. Das Register des Soundchips, das uns interessiert, ist das vierzehnte. Darin liegt der Port A des Chips:

Belegung von Port A des Soundchips

Bit	Bedeutung
0	Seitenauswahl bei der Floppy
1	Auswahlsignal für Laufwerk A
2	Auswahlsignal für Laufwerk B

Die restlichen Bits interessieren uns nicht. Bevor wir also den Floppycontroller selbst ansprechen (bzw. den DMA-Controller), müssen wir erst im Soundchip Seite und Laufwerk einstellen. Auch das finden Sie in einem Programm des Floppykurses, in SELECT.S.

So, jetzt reicht's aber wirklich mit Adressen und Bits und Kommandos und derlei Verwirrendem mehr. Im Grunde habe ich das alles nur für diejenigen Verwegenen herausgearbeitet,

die sich selbst an die Programmierung machen wollen, obwohl ich doch in dieser Ausgabe der ST eine Sammlung von Routinen vorstelle, die jeder abkupfern und zum Vorbild nehmen kann. Natürlich können Sie sich auch noch selbst bemühen; dazu gibt es im nächsten Teil des Floppykurses relativ komplette Informationen über den Floppycontroller (die hätten diesmal alle Heftgrenzen gesprengt).

FREIHEIT FÜR DEN LESEKOPF!

Die Floppyroutinen finden Sie in Listing 1 und 2. Listing 1, LOCKSLEY.S, ist ein Assemblerprogramm, dem der Befehlscode des Controllerbefehls und einige Parameter in einem Feld übergeben werden und das daraufhin selbsttätig die richtige Routine auswählt und sich mit DMA, MFP

FRAGEN ÜBER FRAGEN

Seit der ersten Folge des Floppykurses rennt man mir telefonisch und brieflich die Bude ein. Das ist auch ganz in Ordnung so, denn dabei kann ich ja auch noch lernen. Nur häufen sich gewisse Fragen, und ich nehme an, daß sie von allgemeinem Interesse sind:

Frage 1: Kann man mit HYPERFORMAT auch Festplatten formatieren? Die Antwort: Nein, nein, nein. Erstens nimmt HYPERFORMAT den Floppycontroller in die Mangel und nicht den Festplattencontroller. Zudem ist der bei den verschiedenen Festplatten, die für den ST angeboten werden, auch noch unterschiedlich. Und drittens wäre mir auf der Festplatte Datensicherheit wirklich wichtiger als 20 oder 30 Prozent mehr Kapazität, finden Sie nicht?

Frage 2: Gibt es neue Versionen von HYPERFORMAT und wo bekomme ich sie? Ich arbeite immer noch recht häufig an HYPERFORMAT, weil mir immer wieder etwas dazu einfällt. Die aktuelle Version ist jetzt (Ende Juni) 2.2 mit Tendenz zur 2.3. Sie hat eine zusätzliche Verifyoption und ein paar kleinere Verbesserungen mitbekommen. Zu bekommen ist die jeweils neueste Version mit einigen Extras gegen frankierten Rückumschlag und Diskette sowie 10 DM unter meiner Adresse, die Sie in den Listings finden.

Frage 3: Wie kompatibel sind HYPERFORMAT-Disketten zu anderen Rechnern? Knifflig. Ich könnte mir gut vorstellen, daß Sie HYPERFORMAT auf vielen anderen Laufwerken physikalisch lesen können. Das hängt sehr stark vom Floppycontroller ab, aber da die meisten nach dem MFM-Standard arbeiten, habe ich da eine gewisse Hoffnung. Das logische Format ist natürlich je nach Rechner völlig verschieden. Am ähnlichsten sind Disketten, die unter MS-DOS formatiert wurden. Insofern besteht Hoffnung, falls Atari doch einmal in ferner Zukunft den MS-DOS-Emulator bringt. Vielleicht vertiefe ich mich dann auch noch einmal in die Materie und produziere ein HYPERFORMAT für MS-DOS-Rechner... Wo wir gerade bei Kompatibilitäten sind: Für das Blitter-TOS werde ich eine neue Version von HYPERFORMAT herausbringen. Wie sich HYPERFORMAT mit dem AMIGA verträgt, konnten Sie ja schon in der letzten Folge erfahren.

Frage 4: HYPERFORMAT produziert bei mir Disketten, auf die ich nicht mehr schreiben kann. Tja, das kann viele Ursachen haben. Erstens sollten Sie lieber zweimal nachprüfen, ob Sie auch eine lauffähige Version haben, das heißt, ob Sie nicht vielleicht doch einem Tippfehler erlegen sind oder etwas beim Kopieren schiefgegangen ist. Dann sollten Sie sicher sein, daß Sie HYPERFORMAT auch richtig bedienen (siehe ST 6/87). Und läuft es dann noch nicht, gibt es immer noch mehrere Möglichkeiten: Verwenden Sie schlechte Disketten? Prüfen Sie mal nach, ob mit hochwertigen Disketten nicht was zu machen ist. Dreht Ihr Laufwerk zu schnell? Normal ist eine Umdrehungszahl zwischen 300 und 304 Umdrehungen pro Minute (herauszufinden mit dem Public-domain-Programm SPEED.TOS oder auch mit dem neuen COPYSTAR).

Trifft das alles nicht zu, dann gehören Sie wahrscheinlich zu den vom Schicksal benachteiligten ST-Besitzern, deren Controller HYPERFORMAT (noch) nicht verträgt. Aber selbst dann ist noch nicht aller Tage Abend. Oft hilft es, wenn Sie nicht alle Tracks formatieren, die Sie erreichen können, sondern nur bis Spur 80. Wenn Sie ein Sourcelisting haben, vorzugsweise das der Version 2.1, versuchen Sie die Lückenbytes zu ändern (in der Version 2.1 ganz am Anfang des Listings als gap1 bis gap4 durchparametriert), das Programm zu assemblieren und damit zu formatieren. Wenn Sie das nicht können, fragen Sie einen Freund oder schreiben Sie mir. Wenn Sie selbst Probleme mit HYPERFORMAT hatten und sie irgendwie gelöst haben, lassen Sie es doch die Welt wissen: Schreiben Sie mir oder der „ST-Computer“ über Ihre Erfahrungen – geben Sie dabei bitte genau Ihre Systemkonfiguration an (und auch bitte das Kaufdatum Ihres ST, ich vermute, daß es verschiedene Baureihen gibt). Leider kann ich nicht ausschließen, daß trotz aller Bemühungen einige ST-Besitzer auf HYPERFORMAT verzichten müssen – das Programm geht nun mal bis an physikalische Grenzen heran.

Frage 5: Was kommt als nächstes? Während des Kurses kommen mir immer mehr Ideen zum Thema. Darunter ist das bereits angekündigte Kopierprogramm für HYPERFORMAT-Disketten, ein wirklich superschnelles Formatierprogramm für „normale“ 9- und 10-Sektordisketten, eine ganz besondere RAM-Disk, der Brückenschlag zwischen AMIGA und ST und und und...

Bleiben Sie also dran. In der nächsten Folge werden wir uns noch einmal dem Controller und dessen Programmierung widmen. Der MINIMON wird wohl noch ein wenig wachsen (irgendwann werden wir ihn umtaufen müssen), und schließlich gibt es noch ein paar Anwendungen für den Trackmonitor. Und den Rest behalte ich vorerst für mich. Bis bald!

und FDC herumschlägt. Listing 2, SELECT.S, dient bislang nur dazu, das richtige Laufwerk und die richtige Seite anzuwählen, wird aber noch erweitert werden.

Ein paar Anregungen: Lesen Sie mal Spuren von verschiedenen Disketten ein (normal formatierte und HYPERFORMATierte). Sehen Sie den Unterschied? Bei HYPERFORMAT liegen die Sektoren viel dichter beieinander. Versuchen Sie doch mal, ein eigenes Trackformat zu erzeugen. Wie wäre es mit 16 Sektoren zu 256 Bytes (damit ist das erzeugte Format kompatibel zu bestimmten HP-Rechnern)? Oder mit 1024-Byte-Sektoren?

Schauen Sie sich auch mal – sofern Sie so etwas besitzen – eine kopiergeschützte Diskette an und versuchen Sie herauszufinden, worin der Schutz besteht (1024-Byte-Sektoren, wild durcheinandergewürfelte Sektorgrößen, Checksummen- und andere Fehler...); das sollte übrigens keine Aufforderung zum Raubkopieren sein, deswegen halte ich mich mit konkreten Tips zu bestimmten Programmen auch lieber zurück. Lesen Sie zur Analyse nicht nur den Track – der Read-Track-Befehl des Controllers ist nicht besonders zuverlässig. Verwenden Sie auch Read-Address. Gehen Sie am besten so vor: Ermitteln Sie mit Read-Address, wieviele Sektoren überhaupt vorhanden sind und wie groß diese sind. Dann lesen Sie die ganze Spur ein und vergleichen die Ergebnisse (dazu kann man in MINIMON den Drucker verwenden). Und zuguterletzt schauen Sie sich die einzelnen Sektoren an. Danach müßten Sie ziemlich genau wissen, wie die Spur aussieht, auch wenn der Read-Track-Befehl nicht so funktioniert, wie er das anständigerweise tun müßte.

LOCKSLEY, DER BEFREIER DES LESEKOPFES

Wie sein Namensgeber (Robin of Locksley, im Volksmund Robin Hood) kämpft LOCKSLEY.S für die Freiheit von Unterdrückten, in diesem Fall von repressiv programmierten Leseköpfen unzähliger Laufwerke (Schrittmotoren aller Länder, vereinigt Euch!). LOCKSLEY.S ist einerseits eine Sammlung von Routinen, die Sie immer wieder brauchen werden, wenn Sie sich mit der direkten Floppyprogrammierung, entschlackt von der Last des Betriebssystems, befassen, und andererseits ein Modul, das wie SELECT.S vollständig relokatibel ist. SELECT.S wie LOCKSLEY.S sind mit dem AS68 assembliert worden. Wie man das macht? Dazu ruft man den AS68 so auf: 'AS68 – l –u filename.s'. Der Linker produziert dann nach 'linker filename.prg=filename.o' ein lauffähiges Programm.

Auch bei LOCKSLEY.S finden Sie am Anfang des Programms eine Tabelle von Ein-/Ausgabeparametern, die sich auch bei eventuellen Erweiterungen des Programms nicht verschieben wird. Wenn Sie LOCKSLEY.S benutzen, erwartet das Programm mindestens den Opcode desjenigen Befehls, den der Controller ausführen soll, im Parameter 'opcode'. LOCKSLEY analysiert diesen Opcode und springt dann die richtige Routine für jeden Befehl an. Zu den einzelnen Befehlen können noch weitere Parameter benötigt werden. Eine Übersicht der möglichen Befehle des Controllers und der Ein/Ausgabeparameter finden Sie in der Tabelle auf der nächsten Seite. Was die einzelnen Befehle tun, finden Sie kurz in der Anleitung zum EXTENDED MINIMON erläutert (genaues in der nächsten Folge).

ENE MENE MISTE – DIE QUAL DER WAHL

SELECT.S ist ein Programm, das sich (bis jetzt) ausschließlich damit beschäftigt, die richtige Seite und das richtige Laufwerk auszuwählen. Man ruft es auf, bevor man einen Controllerbefehl abschickt, damit man sicher ist, daß der Formatierbefehl, den man im Sinne hat, auch tatsächlich die freie Disk in Laufwerk B in die Mangel nimmt und nicht die Systemdiskette in Laufwerk A. Versäumt man das, erzeugt das graue Haare, aber wem unter ihnen erzähle ich damit etwas Neues...

Im Quelltext erkennen Sie, daß ganz am Anfang des Programms bestimmte Speicherstellen für die Parameterübergabe durch BASIC (oder andere Sprachen) vorgesehen sind. In 'laufwerk' schreibt man die Bitkombination für aktuelle Seite und gewünschtes Laufwerk (dazu schauen Sie sich am besten die Belegung des Port A im Soundchip an, die im Artikel erwähnt ist). Laufwerk A, Vorderseite wählt man an, indem man „2" in 'laufwerk' schreibt und dann in die Routine springt.

Wenn Sie sich den Quelltext anschauen, werden Sie sich wundern, daß während des Programms alle Interrupts ausgeschaltet werden. Das hat einen einfachen Grund. In einer Interruptroutine prüft das Betriebssystem ständig, ob auf einem Laufwerk eine Diskette gewechselt wurde. Dazu muß es die Laufwerke natürlich selektieren und benutzt dazu den Port A. Damit es da keine Kollisionen gibt, habe ich den Interrupt für diese kurze Zeit einfach verboten.

Ein Wort noch zu einer weiteren Besonderheit. Nach einem Befehl deselektiert man sinnvollerweise das Laufwerk, indem man „0" in 'laufwerk' schreibt und SELECT aufruft. Allerdings kam es dabei vor, daß man das Laufwerk abwählte und trotzdem (oder gerade deswegen) der Laufwerksmotor sich wie das Auto einer großen Marke benahm: Er lief und lief und lief...

Woran das liegt, weiß ich noch nicht genau, ich hoffe da auf die Unterstützung eines kundigen Lesers. Was ich weiß: Wenn man abwartet, bis das Laufwerk seinen Motor von alleine abschaltet, und dann erst die Floppy deselektiert, funktioniert es. Dabei muß man aber in Kauf nehmen, daß nach jedem Befehl eine Sekunde für das Auslaufen des Motors verlorengeht. Diese Warteschleife finden Sie im Programm ab dem Label 'motor'.

In der nächsten Ausgabe werde ich SELECT so erweitern, daß man mit diesem Programm auch die Register des Floppycontrollers direkt von BASIC aus lesen und beschreiben kann. Deswegen sind in das Listing einige Definitionen und „open ends" eingebaut, über die Sie sich vielleicht wundern mögen.

(C.B.)

Die Liste der Routinen:

main	Initialisierung
motor	Warteschleife, bis der Motor ausgelaufen ist
mach_mal	Laufwerk und Seite selektieren
super	Supervisor-/Usermodus einschalten
time	Warteschleife

Für alle ATARI ST

3D-Zeichen-Programm für ATARI ST

GFA OBJEKT

Konstruieren von 3D-Objekten im Baukastenprinzip
Drahtmodelle – Hidden Line – Hidden Surface
Schnittstelle zu GFA-BASIC
Schnittstelle zu GFA-DRAFT plus (Konstruktion)
Schnittstelle zu GFA-VEKTOR (Animation)
Schnittstelle zu Standard-Malprogrammen (Illustration)
Vollständig in GFA-BASIC geschrieben

GFA-OBJEKT DM 198,–

CAD-Programm für ATARI ST

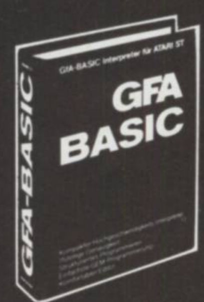
GFA DRAFT plus

Voll GEM-gesteuertes leistungsfähiges CAD-Programm
Maßstabgerechtes Erstellen von Zeichnungen in Zoll und mm
Bildausschnitte und Symbole beliebig manipulierbar und gradweise drehbar
Mächtige Zeichenfunktionen wie z.B. Lot, Winkel zu Geraden, Kreis durch 3 Punkte, etc.
255 Zeichenebenen (10 gleichzeitig darstellbar)
Schraffieren von beliebigen Flächen
Einfachste Handhabung
Läuft auf fast allen Plottern und Druckern
Symbole und Bibliotheken in beliebiger Menge anlegbar

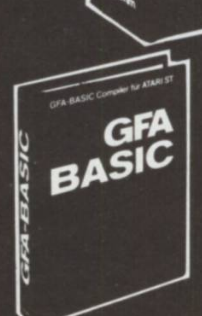
GFA-DRAFT plus ist komfortabler und schneller, mit außergewöhnlichen Features wie:

- Schnittstelle zum GFA-BASIC
- Anbindung an Datenbanken (Stücklistenverwaltung)
- Zeichenfläche bis DIN A0
- Kommandoeingabe auch über Tastatur

GFA-DRAFT plus DM 349,–



GFA-BASIC Interpreter V 2.0
DM 169,–



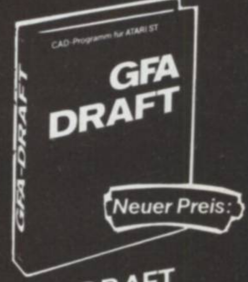
GFA-BASIC Compiler
DM 169,–



GFA-VEKTOR
3D-Grafik-Toolbox zum GFA-BASIC
DM 99,–



GFA-STARTER
Startet GEM-Programme aus dem Autoordner
DM 59,–



GFA-DRAFT
DM 198,–

...Anruf genügt: 0211-588011

GFA-CLUB
GFA-PC-Software
bitte Info anfordern

GFA Systemtechnik GmbH
Heerdter Sandberg 30
D-4000 Düsseldorf 11
Telefon 0211/588011

GFA SYSTEMTECHNIK

Befehl	Parameter
irq	Eingabe: nur Opcode in 'opcode' Ausgabe: keine!
Read Sektor	Eingabe: Opcode in 'opcode', Pufferadresse in 'buffer1', Sektor in 'sector', Anzahl der zu lesenden Bytes in 'count' Ausgabe: gelesene Bytes in 'count', Status des Floppycontrollers in 'fst', DMA-Status in 'dst', Ende der DMA-Übertragung in 'end', Timeoutflag in 'timeout'
Read Track	Eingabe: Opcode in 'opcode', Pufferadresse in 'buffer1', Zahl der zu lesenden Bytes in 'count' Ausgabe: wie Read Sector
Write Track	Eingabe: Opcode in 'opcode', Adresse der Trackdaten in 'buffer1', Zahl der zu schreibenden Bytes in 'count' Ausgabe: geschriebene Bytes in 'count', sonst wie bei Read Sector
Write Sector	Eingabe: Opcode in 'opcode', Adresse der Sektordaten in 'buffer1', Zahl der zu schreibenden Bytes in 'count', Sektornummer in 'sector' Ausgabe: wie bei Write Track
Step, Step-in Step-out, Restore	Eingabe: Opcode in 'opcode' Ausgabe: Controllerstatus in 'fst'
Seek	Eingabe: Opcode in 'opcode', Zielspur in 'track' Ausgabe: wie bei Step etc.
Read Address	Eingabe: Opcode in 'opcode', Zahl der zu lesenden Adreßfelder in 'count', Pufferadresse für die Adreßfelder in 'buffer1', Pufferadresse für die Statusmeldungen in 'buffer2' Ausgabe: wie bei Read Sector

Die Liste der Routinen:

Routine	Beschreibung
main	Initialisiert, verzweigt und macht Schluß
wrfdc	Ein Byte in $ff8604 schreiben (Zugriff auf FDC- oder DMA-Register).
as_time_goes_by	Warteschleife
warten_auf_godot	Wartet auf Ende des FDC-Kommandos
poll	Testet auf Unterbrechungsanforderung
game_over	Zeitüberlauf passiert
fix_und_fertig	DMA-Übertragung am Ende
status	Status des DMA-Chips lesen
dma	Startadresse für DMA setzen
super	Supervisor/Usermodus umschalten
analyze	Analysiert den Opcode und verzweigt
irq	Routine für den FORCE-IRQ-Befehl
rdsec	Routine für den Read-Sector-Befehl
exe	Einsprung für den Zyklus „Kommando schreiben"-„warten"-„Status lesen"
rdtrk	Routine für den Read-Track-Befehl
wrtrk	Routine für den Write-Track-Befehl
wrsec	Routine für den Write-Sector-Befehl
step_dance	Routine für Step, Step-In, Step-Out und Restore
such_hasso	Routine für den Seek-Befehl
rdadr	Routine für den Read-Address-Befehl
nochnid	Schleifenmarke für 'noch 'ne ID lesen'

1

```
LOCKSLEY.S                    Seite 1          (C) 1987 Claus Brod

*********************************************
* LOCKSLEY - Der Befreier des Lesekopfes    *
* Direkter Zugriff auf den Floppycontroller *
*        --- Test auf eigene Gefahr ---     *
* Mit Schnittstelle nach außen für den      *
* Anschluß an MINIMON                       *
* Written 1987 by                           *
*     Claus Brod                            *
*     Am Felsenkeller 2                     *
*     8772 Marktheidenfeld                  *
*     09391/3206                            *
*                                           *
* Version 1.0                               *
*   6.6.87 (starting all over again)        *
*   7.6.87 (main work)                      *
*   8.-11.6.87 (debugging)                  *
*   12.6.87 (adding some sub's)             *
*   13.6.87 (fiddling with #@!?% Read Adr.) *
*   14.-16.6.87 (cleaning up)               *
* Assembliert mit AS68                      *
*********************************************

*************************
* Ein paar Definitionen für den langen Weg
*************************

mfp - $fffa01               * Adresse des MFP68901 fürs Polling

daccess- $ff8604            * DMA-Controller, FDC-Zugriff oder Sektorzähler
dmodus - $ff8606            * DMA-Controller, Modus einstellen
dlow   - $ff860d            * DMA-Controller, Übertragungsstart Lowbyte
dmid   - $ff860b            * DMA-Controller, Übertragungsstart Midbyte
dhigh  - $ff8609            * DMA-Controller, Übertragungsstart Highbyte
time   - $40000             * Timeout-Konstante

*************************
* Der Sprung ins Ungewisse
*************************

  bra main                  * Sprung zum Programmanfang

*************************
* Ein/Ausgabefeld zur Parameterübergabe
*************************

opcode:   .dc.w 0               * Hier kommt der Opcode
*                                 des Controllerbefehls hinein
track:    .dc.w 0               * Tracknummer
*
sector:   .dc.w 0               * Sektornummer
*
count:    .dc.w 0               * in:zu übertragende Bytes/ID-fields etc.
*                                 out:übertragene Bytes
buffer1:  .dc.l 0               * Adresse des ersten Puffers für
*                                 Track-, Sektor-, ID-Daten
buffer2:  .dc.l 0               * Adresse des Reservepuffers
*
fst:      .dc.w 0               * Status des Controllers
*
dst:      .dc.w 0               * Status der DMA
*
```

2

```
LOCKSLEY.S                    Seite 2          (C) 1987 Claus Brod

dstart:   .dc.l 0               * Start der DMA-Ubertragung
*
dend:     .dc.l 0               * Ende der DMA-Ubertragung
*
timeout:  .dc.w 0               * Timeoutflag
*

stk:      .dc.l 0               * Puffer für Stackpointer
*
dflag:    .dc.w 0               * DMA-Flag
*
**************************
* Hauptverteilerroutine
**************************

main:
   movem.l d0-d7/a0-a6,-(sp)    * Register verschwinden lassen
   clr.l d0                     * Userstack wird Supervisorstack
   bsr super                    * Supervisormodus an
   lea stk(pc),a2               * Adresse des Puffers für den Stackpointer
   move.l d0,(a2)               * Stackpointer retten
   move.w opcode(pc),d6         * FDC-Befehl holen
   lea dflag(pc),a2             * Adresse des DMA-Flags
   move.w #0,(a2)               * DMA-Flag initialisieren
   lea timeout(pc),a2           * Adresse des Timeoutflags
   move.w #0,(a2)               * Timeout-Flag initialisieren
   st $43e                      * Floppy-Vertical-Blank-Interrupt sperren
   bsr analyze                  * Befehl analysieren und ausführen
   sf $43e                      * Floppy-VBL wieder erlauben
   lea stk(pc),a2               * Adresse des Stackpointerpuffers
   move.l (a2),d0               * Stackpointer holen
   bsr super                    * Supervisormodus aus (schaaade...)
   movem.l (sp)+,d0-d7/a0-a6    * Register wieder holen
   rts                          * und raus (ade!)

*****************************
* wrfdc: Byte in d7 an den Controller
* schicken
*****************************
wrfdc:
   move.w #30,d1                * Zähler auf 30
   bsr as_time_goes_by          * Warteschleife
   move.w d7,daccess            * d7 ins Access-Register des DMA-Chips
   move.w #30,d1                * Zähler auf 30

*****************************
* as_time_goes_by: Warteschleife
* mit d1 Durchläufen
*****************************
as_time_goes_by:
   dbf d1,as_time_goes_by       * Looping (huiii...)
   rts                          * back to the future

*****************************
* warten_auf_godot : Wartet auf das IRQ-Signal
* des FDC
*****************************
warten_auf_godot:
   move.l #time,d7              * Konstante für Timeout
```

(3)

```
LOCKSLEY.S                 Seite 3          (C) 1987 Claus Brod

poll:
    btst #5,mfp                 * IRQ am MFP?
    beq fix_und_fertig          * jawoll, Kommando ist ausgeführt
    subq.l #1,d7                * Timeoutzähler magert ab
    beq game_over               * Sorry, zu lange gefackelt
    lea dflag(pc),a2            * Adresse des DMA-Flags
    tst.w (a2)                  * DMA aktiv?
    beq poll                    * nein, zum Polling
    lea buffer2(pc),a2          * Adresse der Adresse (!) des Reservepuffers
    move.b dhigh,1(a2)          * Highbyte der DMA-Adresse
    move.b dmid,2(a2)           * Midbyte der DMA-Adresse
    move.b dlow,3(a2)           * Lowbyte der DMA-Adresse
    move.l buffer2(pc),d0       * DMA-Adresse nach d0
    cmp.l dend(pc),d0           * Mit Endadresse vergleichen
    blt poll                    * Noch nicht erreicht, weiter testen

    move.b #$d0,d7              * Force IRQ
    bsr irq                     * FDC unterbrechen
    lea dflag(pc),a2            * Adresse des DMA-Flags
    move.w #0,(a2)              * DMA beendet
    bra fix_und_fertig          * Feierabend

***************************
* Timeout-Konstante ist abgelaufen
***************************
game_over:
    bsr fix_und_fertig          * Status lesen und in fst ablegen
    move.b #$d0,d7              * Force IRQ
    bsr irq                     * FDC unterbrechen
    lea timeout(pc),a2          * Adresse des Timeout-Flags
    move.w #1,(a2)              * Timeout passiert
    lea dflag(pc),a2            * Adresse des DMA-Flags
    move.w #0,(a2)              * DMA völlig am Ende
    rts                         * und fertig

***************************
* DMA-Ubertragung oder FDC fertig
***************************
fix_und_fertig:
    move.w daccess,d0           * Status lesen
    lea fst(pc),a2              * Adresse des FDC-Status-Flags
    move.w d0,(a2)              * Status ablegen
    rts                         * und raus

***************************
* status: Status und Bytezahl lesen
***************************
status:
    move.w dmodus,d0            * Status lesen
    lea dst(pc),a2              * Adresse des DMA-Status-Flags
    move.w d0,(a2)              * Status schreiben
    clr.w d1                    * d1 löschen
    move.b dhigh,d1             * DMA-High
    lsl.l #8,d1                 * um ein Byte schieben
    move.b dmid,d1              * DMA-Mid
    lsl.l #8,d1                 * um ein Byte schieben
    move.b dlow,d1              * DMA-Low
    lea dend(pc),a2             * Adresse der Endadresse
    move.l d1,(a2)              * Endadresse
    sub.l dstart(pc),d1         * minus Start
    lea count(pc),a2            * Adresse des Bytezählers
```

(4)

```
LOCKSLEY.S                 Seite 4          (C) 1987 Claus Brod

***************************
* dma: dma setzen (Spiegelbild zu status)
***************************
dma:
    lea dstart(pc),a2           * Adresse der Startadresse
    move.l d7,(a2)              * Startadresse ablegen
    move.b d7,dlow              * Lowbyte
    lsr.l #8,d7                 * um ein Byte schieben
    move.b d7,dmid              * Midbyte
    lsr.l #8,d7                 * um ein Byte schieben
    move.b d7,dhigh             * Highbyte
    move.l dstart(pc),d7        * Startadresse holen
    clr.l d0                    * d0 löschen
    move.w count(pc),d0         * Bytecounter nach d0
    add.l d0,d7                 * Addieren
    lea dend(pc),a2             * Adresse der Endadresse
    move.l d7,(a2)              * Endadresse ablegen
    rts

***************************
* super: Schaltet vom Usermode
* in der Supervisormode und umgekehrt
* Ubergabe des Stackpointers in d0
***************************
super:
    move.l d0,-(sp)             * Stackpointer auf Stack
    move.w #$20,-(sp)           * SUPER
    trap #1                     * im GEMDOS
    addq.l #6,sp                * Stack korrigieren
    rts

***************************
* analyze: Analysiert grob, welche Art von
* Befehl vorliegt
* und verteilt entsprechend
* in: d6.w Opcode
***************************
analyze:
    move.w d6,d7                * Opcode retten
    move.w d6,d5                * gleich nochmal
    btst #7,d6                  * Bit 7 testen
    bne typii                   * kein Typ-I-Befehl
    and.b #$f0,d6               * obere 4 Bits ausmaskieren
    cmp.b #16,d6                * SEEK-Befehl?
    beq such_hasso              * jawoll
    bne step_dance              * kein Seek, aber Typ-I

typii:
    btst #6,d6                  * Bit 6 testen
    bne typiii                  * kein Typ-II-Befehl
    btst #5,d6                  * Bit 5 testen
    beq rdsector                * gelöscht, also Read-Sector-Befehl
    bne wrsector                * Write-Sector-Befehl

typiii:
    and.b #$f0,d6               * obere 4 Bits ausmaskieren
    cmp.b #$c0,d6               * Read Adress?
    beq rdadr                   * jawoll
    cmp.b #$e0,d6               * Read Track?
    beq rdtrk                   * jawoll
```

Ihr ST auf dem NEUESTEN STAND

DAS PC-GEHÄUSE



KOMPAKT-KIT .... 498,00 DM

- ★ Flaches, abgesetztes TASTATUR-GEHÄUSE mit RESTKNOPF und voll entstörter Schnittstellenplatine und SPIRALKABEL.
- ★ Hauptgehäuse ist vorbereitet für bis zu 2 LAUFWERKE UND EINE HARDDISK (Atari und die meisten Fremdhersteller) mit allen dazu benötigten Kabel, Befestigungen und Blenden.
- ★ SCHALTNETZTEIL (VDE- und Post-zugelassen: versorgt Rechner, Harddisk und Laufwerke. ZENTRALER NETZSCHALTER an der Vorderseite des Hauptgehäuses.
- ★ Hauptgehäuse wird auf ST-Untergehäuse mit Zwischendeck aufgebaut, sodaß ALLE URSPRÜNGLICHEN SCHNITTSTELLEN BLEIBEN. Der komplette Einbau OHNE LÖTEN – AUSFÜHRLICHE GEBRAUCHSANLEITUNG.

KOMPAKT KIT + DOPPELSEITIGES NEC LAUFWERK 728,– DM
(EINBAU IN KIT OHNE ZUSATZTEILE)

HARDDISK-ERWEITERUNGS KIT .... 98,00 DM

- ★ Benötigtes Kabel und Einbaumaterial für Atari Harddisk.
- ★ Zeitverzögerungsschaltung: Gewährleistet gemeinsames Anschalten von Harddisk und Rechner über zentralen Netzschalter.
- ★ Akku-Pufferung für die Uhr innerhalb des Tastaturprozessors (akkus extra).

TASTATURGEHÄUSE . 128,00 DM

Resetknopf, Spiralkabel, entstörte Schnittstellen für Maus & Joystick!

DISKETTENSTATIONEN

NEC 3,5" DOPPELSEITIG Laufwerke in Gehäuse mit Stromversorgung. Voll ATARI kompatibel. Anschlussfertig.

EINZELSTATION ..... 398,00 DM
DOPPELSTATION .... 698,00 DM

SPEZIELL FÜR IHREN ATARI 260/520 ST

KOMPAKT-KIT+
ALLES IN EINEM

★ Das Neue NEC 1036A 3,5" Drive. 1 Mbyte......... 243,00 DM
Aztec Schaltnetzteile (5v 4A, 12v 2A, – 12v 0,2A) ... 118,00 DM ★

Riedstr. 2 · 7100 Heilbronn
TEL. 0 71 31 / 7 84 80

Fa. ANTHONY SEXTON
Technische Entwicklung

Die Kunst der Software.

Seit gut eineinhalb Jahren vertreiben wir Software für den Atari ST. **Signum!, STAD** und der **Megamax C – Compiler** sind seitdem einer immer größer werdenden Anzahl von Anwendern ein Begriff geworden. Bei der Herstellung und dem Verkauf von hochwertiger Software kann Qualität nicht nur aus Worten bestehen, denn Fehlerhaftes oder gar Halbheiten werden vom Benutzer nachhaltig gemerkt. Unser Qualitätsgedanke liegt daher in einer zwingenden Abfolge guter Programmideen und deren perfekter Durchführung begründet, sowie einer intensiven Kommunikation mit den Programmautoren und andererseits: den Anwendern. Die Vorbereitung eines Programmes dauert nicht selten ein ganzes Jahr, bevor es veröffentlicht werden kann, deshalb können wir gute, ausgereifte Software auch nicht massenhaft auf den Markt bringen, sondern eher ein, höchstens zwei Programme jährlich neu ankündigen. Denn Sorgfalt und Service brauchen Zeit. Obwohl wir ein junges Team sind, sehen wir – jeder für sich – auf eine langjährige Erfahrung in der Branche zurück. Müssen wir auch, denn der fast größte Anteil unserer Arbeit gilt der Programmbetreuung nach einer Veröffentlichung. Dann fangen die Telefone an zu klingeln, Ratschläge und Starthilfen werden erteilt. Fragen, Antworten, Glückwünsche. Wir finden, daß dieser Aufwand sein muß, denn eine vernünftige Software wird ständig verändert, erneuert, verbessert und ergänzt. Damit haben unsere registrierten Anwender natürlich den Vorteil aktueller Programme. Das Feedback eben dieser Anwender hat uns bestätigt.



Herzlichst,
Application Systems Heidelberg

APPLICATION SYSTEMS HEIDELBERG

Bitte senden Sie mir ausführliches Prospektmaterial zu Ihren Softwareprodukten.

ST987

Name, Vorname ____________

Adresse ____________

____

Telefon ____________

Branche, Beruf ____________

Meine Rechnerkonfiguration ____________

____

Ich benutze meinen Atari vorwiegend beruflich / privat.

Obenstehenden Coupon bitte einsenden an APPLICATION SYSTEMS HEIDELBERG, Englerstraße 3, 6900 Heidelberg. Noch Fragen? Telefon (06221) 300002.

(5)

```
LOCKSLEY.S                    Seite 5         (C) 1987 Claus Brod

*                               * alles andere ist Force Interrupt

**************************
* irq: Unterbricht den Controller
**************************
irq:
   bsr wrfdc                    * d7 an den Controller
   move.w #250,d1               * 250 Schleifendurchläufe
   bra as_time_goes_by          * kurz warten

*****************************
* rdsector: Liest einen Sektor oder
* gleich einen ganzen Haufen davon
*****************************
rdsector:
   move.l buffer1(pc),d7        * Pufferadresse
   bsr dma                      * als DMA-Adresse
   lea dflag(pc),a2             * Adresse des DMA-Flags
   move.w #1,(a2)               * DMA-Flag setzen
   move.w #$90,dmodus           * mit der Schreib/Leseleitung
   move.w #$190,dmodus          * kippeln, löscht den DMA-Status
   move.w #$90,dmodus           * Sektorzähler des DMA-Chips
   move.w #14,d7                * maximal 14 Sektoren (utopisch)
   bsr wrfdc                    * d7 an FDC
   move.w #$84,dmodus           * Sektorregister
   move.w sector(pc),d7         * Sektornummer
   bsr wrfdc                    * d7 an FDC
   move.w #$80,dmodus           * Kommandoregister
exe:
   move.w d5,d7                 * Kommando von d5 nach d7
   bsr wrfdc                    * d7 an FDC
   bsr warten_auf_godot         * Ende des Kommandos abwarten
   bra status                   * Status lesen

*****************************
* rdtrk: Track lesen
*****************************
rdtrk:
   move.l buffer1(pc),d7        * Adresse des Spurpuffers
   bsr dma                      * DMA initialisieren
   lea dflag(pc),a2             * Adresse des DMA-Flags
   move.w #1,(a2)               * DMA in Arbeit!
   move.w #$90,dmodus           * Schreib/Leseleitung
   move.w #$190,dmodus          * umschalten, siehe oben
   move.w #$90,dmodus           * Sektorzähler des DMA-Chips
   move.w #14,d7                * 14 Sektoren
   bsr wrfdc                    * d7 an FDC
   move.w #$80,dmodus           * Kommandoregister
   bra exe                      * Kommando schreiben und beenden

*****************************
* wrtrk: Track schreiben
*****************************
wrtrk:
   move.l buffer1(pc),d7        * Spurpufferadresse nach d7
   bsr dma                      * DMA initialisieren
   lea dflag(pc),a2             * Adresse des DMA-Flags
   move.w #1,(a2)               * DMA in Arbeit!
   move.w #$190,dmodus          * Schreib/Leseleitung umschalten
   move.w #$90,dmodus           * (bewirkt was? Raten Sie mal!)
```

(6)

```
   bsr wrfdc                    * d7 an FDC
   move.w #$180,dmodus          * Kommandoregister wählen
   bra exe                      * Kommando schicken und beenden

*****************************
* wrsector: Sektor schreiben
*****************************
wrsector:
   move.l buffer1(pc),d7        * Adresse des Puffers
   bsr dma                      * DMA initialisieren
   lea dflag(pc),a2             * Adresse des DMA-Flags
   move.w #1,(a2)               * DMA in Arbeit
   move.w #$190,dmodus          * mit der Schreib/Leseleitung
   move.w #$90,dmodus           * klappern
   move.w #$190,dmodus          * Sektorzähler des DMA-Chips
   move.w #14,d7                * Maximal 14 Sektoren (utopisch)
   bsr wrfdc                    * d7 an FDC
   move.w #$184,dmodus          * Sektorregister des FDC
   move.w sector(pc),d7         * Startsektor holen
   bsr wrfdc                    * d7 an FDC
   move.w #$180,dmodus          * Kommandoregister des FDC
   bra exe                      * Kommando schicken und beenden

*****************************
* step_dance: einheitliche Routine für step,
* step-in, step-out, restore
*****************************
step_dance:
   move.w #$80,dmodus           * Command-Register auswählen
   bsr wrfdc                    * Byte in d7 an FDC
   bra warten_auf_godot         * Auf FDC warten...

*****************************
* such_hasso: Seek-Befehl
*****************************
such_hasso:
   move.w #$86,dmodus           * Datenregister auswählen
   move.w track(pc),d7          * Tracknummer holen
   bsr wrfdc                    * Byte in d7 an FDC
   move.w #$80,dmodus           * Commandregister
   move.w d5,d7                 * Kommando
   bsr wrfdc                    * Byte an FDC
   bra warten_auf_godot         * Auf FDC warten

*****************************
* rdadr: Adreßfeld lesen
*****************************
rdadr:
   move.l buffer1(pc),d7        * Adresse des Puffers holen
   move.l buffer2(pc),a3        * Adresse des Reservepuffers
   bsr dma                      * DMA initialisieren
   move.w #$90,dmodus           * Schreib/Leseleitung
   move.w #$190,dmodus          * umschalten
   move.w #$90,dmodus           * und auf Sektorzähler
   move.w #1,d7                 * Maximal 1 Sektor
   bsr wrfdc                    * d7 an FDC (DMA)
   move.w #$80,dmodus           * Kommandoregister
   move.w count(pc),d2          * Wieviele ID-Felder?

nochnid:
   move.w d5,d7                 * Kommando holen
   bsr wrfdc                    * und schreiben
   bsr warten_auf_godot         * auf FDC warten
   move.w fst(pc),d1            * FDC-Status holen
   move.w d1,(a3)+              * Status in den zweiten Puffer
   dbra d2,nochnid              * Noch eine ID
   bra status                   * und zum Status
```

⑦

```
SELECT.S                      Seite 1          (C) 1987 Claus Brod
****************************
* SELECT.S
* Selektiert Seite und Laufwerk
* Mit Schnittstelle für BASIC
* (C) 1987 and for all eternity by
*       Claus Brod
*       Am Felsenkeller 2
*       8772 Marktheidenfeld
*       09391/3206
* Version 1.0
* Last update 16.6.87
* Assembliert mit AS68
*****************************

*****************************
* Definitionen
*****************************
snd = $ff8800              * Adresse des Soundchips
sndwrt = $ff8802           * Ein/Ausgabe des Soundchips
dmodus = $ff8606           * DMA-Modus-Register
daccess = $ff8604          * DMA-Access

*****************************
* Sprung ins Hauptprogramm
*****************************
   bra main
*****************************
* Eingabefeld für BASIC
*****************************
laufwerk: .dc.w 0          * Eingabeparameter Laufwerk und Seite
parm: .dc.l 0              * Reserveein/ausgabefeld für Erweiterungen
temp: .dc.l 0              * Reservefeld
stk:  .dc.l 0              * Puffer für Stackpointer

*****************************
* Und jetzt das Hauptprogramm
*****************************
main:
   movem.l d0-d7/a0-a6,-(sp)     * Register retten
   clr.l d0                      * Userstack wird Supervisorstack
   bsr super                     * Supervisormode an
   lea stk(pc),a2                * Stackpointer
   move.l d0,(a2)                * retten

   st $43e                       * Floppy-VBL sperren
   move.w laufwerk(pc),d7        * Laufwerksnummer holen
   bne mach_mal                  * Wenn nicht gerade 0, dann mach los
   move.w #$80,dmodus            * Statusregister
motor:
   move.w daccess,d1             * auslesen
   btst #7,d1                    * Motor noch an?
   bne motor                     * jawoll
mach_mal:
   cmpi.b #8,d7                  * Laufwerkscode größer als 7?
   bge extend                    * ja, extended mode
   eor.b #7,d7                   * Bits invertieren
   and.b #7,d7                   * und ausmaskieren
   move.w sr,-(sp)               * Status retten
   or.w #$700,sr                 * Interrupts aus
   move.b #14,snd                * Port A wählen
   move.b snd,d0                 * Port A lesen
   and.b #$f8,d0                 * ausmaskieren
   or.b d0,d7                    * neue Seite/neues Laufwerk setzen
   move.b d7,sndwrt              * in Port A
```

⑧

```
SELECT.S                      Seite 2          (C) 1987 Claus Brod
   sf $43e                       * Floppy-VBL erlauben
   move.w (sp)+,sr               * Status wieder holen
raushier:
   lea stk(pc),a2                * Stackpointer
   move.l (a2),d0                * zurückholen
   bsr super                     * wieder in den Usermode
   movem.l (sp)+,d0-d7/a0-a6     * Register zurückholen
   rts                           * und raus

**************************
* super: Schaltet vom Usermode
* in den Supervisormode und umgekehrt
* in: d0 Stackpointer
**************************
super:
   move.l d0,-(sp)               * Stackpointer auf Stack
   move.w #$20,-(sp)             * SUPER
   trap #1                       * im GEMDOS
   addq.l #6,sp                  * Stack reparieren
   rts                           * und raus

**************************
* time: Warteschleife
**************************
time:
   dbra d1,time                  * Zähler magert ab
   rts                           * und raus

**************************
* msg: Gibt String ab a2 aus
**************************
msg:
   move.l a2,-(sp)               * Adresse des Strings auf den Stack
   move.w #9,-(sp)               * PRINT LINE
   trap #1                       * im GEMDOS
   addq.l #6,sp                  * Stack korrigieren
   rts                           * und raus

**************************
* extend: Wird in der nächsten ST ergänzt
**************************
extend:
   lea nochnit(pc),a2            * extend noch nicht
   bsr msg                       * implementiert
   bra raushier                  * und raus

nochnit:
.dc.b 'Routine noch nicht implementiert.',13,10,0
```

Listing 1: Do__file in C

```c
#include <gemdefs.h>
#include <obdefs.h>
#include <osbind.h>

char path[64],name[64],sr[10],ret[64];

main()
{
   int c,taste,form_key=1;

   appl_init();
   graf_mouse(ARROW,0l);

   strcpy(sr,"*.*");       /* Kann gegen beliebig ausgetauscht werden */
   path[0]=0;
   while (form_key==1)
   {
      taste=do_file(path,name,sr,ret);
      if (taste==0)
         form_key=form_alert(1,"[1][Taste ABBRUCH gedrückt][ OK | ABBRUCH ]");

      else
         form_key=form_alert(1,"[1][Taste OK gedrückt][ OK | ABBRUCH]");
   }
   appl_exit();
}

/********************************/
/*                              */
/*      Pfad-Handhabung         */
/*            mit               */
/*      Fileselektorbox         */
/*                              */
/*   14.07.1987  S. Hähn        */
/*                              */
/********************************/

/* fpath[] = Pfadname:  fpath[0]=0 bedeutet Einsetzen des aktuellen
                        Pfades

   fname[] = Dateiname: fname erscheint in der Fileselektorbox und
                        ist getrennt von dem eigentlichen Pfadnamen

   srch[]  = Extender:  mögliche Pfaderweiterung wie zum Beispiel
                        '*.*', '*.pas', '*.c' etc.

   fpn[] = kompletter Pfad:  enthält Pfad wie auch den Dateiname

   Rückgabewert =    0:  Die Taste ABBRUCH wurde gewählt
                     1:  Die Taste   OK    wurde gewählt

*/

int do_file(fpath,fname,srch,fpn)
char *fpath,*fname,*srch,*fpn;

{
   char tpath[64];
   char tname[64];
   int i;
   int key;
   long str_len();

   strcpy(tpath,fpath);                 /* alten Pfad retten */
   strcpy(tname,fname);                 /* alten Dateinamen retten */
   if(fpath[0] == '\0')                 /* wenn kein Pfad vorhanden */
      {                                 /* aktuellen Pfad erstellen */
      fpath[0] = Dgetdrv() + 'A';       /* Laufwerksbezeichnung holen */
      strcpy(&fpath[1],":");            /* Doppelpunkt anhängen */
      Dgetpath(fpn,Dgetdrv());          /* Pfad holen */
      strcat(fpath,fpn);                /* Pfad anhängen */
      strcat(fpath,"\\");               /* Backslash anhängen */
      }
   strcat(fpath,srch);                  /* Suchpfad anhängen */
   fsel_input(fpath,fname,&key);        /* Fileselektorbox aufrufen */
   if(key == 1)                         /* OK gedrückt */
      {                                 /* Pfad und Namen zusammenkopieren */
      i = strlen(fpath);                /* Ende des Gesamtpfades */
      while(fpath[i] != 0x5c)           /* Rücksuchen bis Backslash */
          i--;                          /* also Anfang des Suchpfades */
      fpath[++i] = '\0';                /* abschneiden */
      strcpy(fpn,fpath);                /* Pfad kopieren */
      strcat(fpn,fname);                /* Namen anhängen */
      }
   else
      {
      strcpy(fpath,tpath);              /* Pfad wieder zurückkopieren */
      strcpy(fname,tname);              /* Namen zurückkopieren */
      strcpy(fpn,"\0");                 /* Gesamtpfad auf Null setzen */
      }
   return(key);                         /* Taste zurückgeben */
}
```



Auf dem richtigen Pfade

In diesem Monat wollen wir uns ein wenig mit der Pfaderkennung von Programmen beschäftigen, da dies besonders im Zusammenhang mit der Fileselectorbox gewisse Schwierigkeiten zu bereiten scheint. Außerdem wollen wir auf den Objekttyp ICON eingehen, aus dem viel mehr herauszuholen ist, als mancher Programmierer denkt.

Zunächst wenden wir uns aber der Pfaderkennung zu. Dazu schauen wir uns noch einmal kurz den Diskettenaufbau an. Auf jeder Diskette können eine gewisse Anzahl von Dateien abgespeichert werden. Um bei einer grossen Anzahl von Dateien eine bessere Übersicht zu gewährleisten und die Zeit des Suchens zu verkürzen, besteht zusätzlich die Möglichkeit, die Diskette in Ordner einzuteilen, in denen sich wiederum Dateien befinden. Gerade auf einer Festplatte, bei der dreihundert Dateien keine Seltenheit sind, hilft dies, die Übersicht zu gewährleisten. Beispielsweise würde man auf der obersten Ebene ausschließlich Ordner wie SPRACHEN, MALPROGRAMM, UTILITIES anlegen. In dem Ordner SPRACHEN könnten dann wiederum Ordner mit den Namen C, PASCAL, BASIC etc. liegen, in denen die entsprechenden Programme zu finden sind.

Bild 1:

Pfadname: A:\ordner ...\search.src

Diskstation — die Ordner — Suchkr.

"Aufbau eines Pfades "

Listing 2: Do_file in Pascal

```
PROGRAM Test;
CONST {$I GEMCONST.PAS}

TYPE  {$I GEMTYPE.PAS}
      pathname=packed Array [1..80] of Char;

VAR pfad,name,raus,a,such:String;
    x:Boolean;

{$I GEMSUBS.PAS}

FUNCTION dgetdrv:integer;
GEMDOS($19);

PROCEDURE dgetpath(VAR path:pathname;drive:integer);
GEMDOS($47);

FUNCTION Do_File(VAR fpath,fname,srch,fpn:String):Boolean;
VAR tpath,tname:String;
    mpath:pathname;
    i,f:integer;
    key:boolean;
Begin
 tpath:=fpath;                    {Pfadname retten}
 tname:=fname;                    {Programmname retten}
 If fpath='' Then                 {bei leerem Pfad muß der aktuelle Pfad}
  Begin                           {genommen werden}
   fpath:=concat(chr(dgetdrv+ord('A')),':'); {Diskette und :}
   dgetpath(mpath,dgetdrv);       { aktuellen Pfad holen }
   f:=1;
   While mpath[f]<>chr(0) Do      { ankopieren }
    Begin
     fpath[f+2]:=mpath[f];
     f:=f+1;
    End;
   fpath[f+2]:='\';               { Backslash anhängen }
   fpath[0]:=chr(f+2);            { Länge des Strings setzen }
  End;
 f:=length(fpath);
 While fpath[f]<>'\' Do f:=f-1;
 fpath[0]:=chr(f);
 fpath:=concat(fpath,srch);
 key:=Get_In_File(fpath,fname);  {Aufruf der File-Selector-Box}
 If key Then                      {wenn mit "OK" bestätigt wurde}
  Begin                           {muß der neue Pfad und der neue Name}
   fpn:=fname;                    {genommen werden}
   f:=length(fpath);
   While fpath[f]<>'\' Do f:=f-1;
   fpath[0]:=chr(f);
   f:=length(fname);
   While fname[f]<>'\' Do f:=f-1;
   fname:=copy(fname,f+1,length(fname)-f);
  End
 Else                             {ansonsten werden die alten Daten genommen}

  Begin
   fpath:=tname;
   fname:=tname;
  End;

 Do_File:=key;                    {Ergebnis der Funktion wird festgesetzt}
End;

BEGIN
 If Init_Gem>=0 Then
  Begin
   writeln('Eingabe: ');
   write('Pfad: '); readln(pfad);
   write('Name: '); readln(Name);
   write('Such: '); readln(Such);
   x:=Do_File(pfad,name,such,raus);
   writeln('Ausgabe ');
   writeln('Pfad -> ',pfad);
   writeln('Name -> ',name);
   writeln('Raus -> ',raus);
   writeln('Ergebnis ->',x);
   readln(a);
   Exit_Gem;
  End;
END.
```

Wir sehen, daß zu einem Programmnamen nicht nur der eigentliche Dateiname, sondern auch die Diskettenstation und der Pfad – dies sind die Namen der einzelnen Ordner – gehören. Ein Beispiel: Wollten wir das Programm TURBO.PRG aufrufen, das in dem Ordner PASCAL im Ordner SPRACHEN auf der Diskettenstation B liegt, so müßten wir folgenden Gesamtpfad angeben: „B:\SPRACHEN\PASCAL\TURBO.PRG". Auffällig dabei ist, daß jeder Ordner durch das Zeichen '\' (Backslash) getrennt werden muß. Mit jedem weiteren '\' kommt man in der Verschachtelung eine Ebene tiefer.

Nun wollen wir die ganze Theorie etwas veranschaulichen, uns den im Betriebssystem zur Verfügung stehenden Routinen zuwenden und dabei eine C-Routine erarbeiten, die uns für die Pfadverwaltung die größte Arbeit abnimmt. Konstruieren wir uns eine Routine, die bei nicht vorhandenem Pfad den momentan aktuellen Pfad und die dazugehörige Diskettenstation heraussucht, die Fileselectorbox aufruft und dann den selektierten Namen mit dem eingestellten Pfadnamen verknüpft. Diese Routine heißt do_file() und ist in den Listings 1 (C), 2 (PASCAL) und 3 (BASIC) zu finden. Am C-Listing möchte ich nun die uns zur Verfügung stehenden Routinen erarbeiten.

Als Übergabe benötigt do_file() den Pfadnamen, den man auch einfach auf Null setzen kann, um den aktuellen Pfadnamen eintragen zu lassen; denjenigen Dateinamen, der, falls vorhanden, als Vorgabe in der Fileselectorbox erscheint und einen zusätzlichen Extender wie '.pas' oder '★.★' etc. hat, um nach bestimmten Dateien suchen zu können. Wenden wir uns also dem Ablauf der Prozedur zu.

Zunächst werden angegebener Pfad- und Dateiname in den Feldern tpath[] und tname[] zwischengespeichert. Dann wird geprüft, ob ein Pfadname angegeben wurde oder nicht. Ist kein Pfad angegeben, so muß die Prozedur den aktuellen Pfad ermitteln.

Ermitteln des aktuellen Pfades

Der Gesamtpfad ist, wie in Bild 1 gut zu erkennen ist, aus verschiedenen Teilen zusammengesetzt: Laufwerksbezeichnung, Doppelpunkt, Ordner und Wildcard (Suchangabe wie zum Bei-

spiel '★.pas'). Der erste Schritt, um den Gesamtpfad zu erstellen, ist das Ermitteln des aktuellen Laufwerkes. Dies geschieht durch die Routine Dgetdrv(), die sich im GEMDOS befindet (#define Dgetdrv() gemdos(0x19)). Diese Routine liefert einen Wert – angefangen bei Null – zurück, der das entsprechende Laufwerk kennzeichnet. Dabei entspricht eine Null dem Laufwerk A, eine 1 dem Laufwerk B usw. Addieren wir nun noch den ASCII-Wert des Buchstabens A ('A') hinzu, erhalten wir die Laufwerkskennung als Buchstaben. Anschließend wird an den Buchstaben ein Doppelpunkt angehängt. Als weiteren Schritt müssen wir den aktuellen Pfad ermitteln. Auch in diesem Fall ist uns GEMDOS ein eifriger Helfer. Dgetpath() benötigt als Parameter einen String, in den der Pfad hineingeschrieben wird – in unserem Fall fpn[] – und das gewünschte Laufwerk, das wir oben schon herausgefunden haben. Diesen gewonnenen Pfad hängen wir an die Laufwerksbezeichnung an und schließen ihn mit einem Backslash '\' vorerst ab. Als letzten Arbeitsgang vor dem Aufruf der Fileselectorbox, wird noch der Suchpfad '★.pas' angefügt. Wichtig ist, daß, wenn keine bestimmten Dateien gewünscht sind, der Suchpfad '★.★' angehängt wird (dies könnte auf Wunsch in die do__file()-Routine eingebaut werden, zum Beispiel durch 'if (!srch[0]) then strcat(fpath,"★.★");').

Nun wird die Fileselectorbox aufgerufen, in der unser Gesamtpfad in der oberen Zeile und unser Dateiname – sofern vorhanden – in der rechten Zeile erscheint. Nach dem Verlassen der Box durch Anklicken der OK- oder ABBRUCH-Taste, müssen wir eine Fallunterscheidung treffen. Hat der Benutzer die Taste ABBRUCH angewählt, so werden die alten Bedingungen wieder hergestellt, indem der zwischengespeicherte Pfadname und Dateiname in die ursprünglichen Variablen kopiert werden. Außerdem wird der Rückgabestring gelöscht, da kein sinnvoller Zusammenhang zwischen eventuell angewähltem Pfad und Namen besteht. Hat der Benutzer aber die OK-Taste angeklickt, so müssen wir aus dem Pfad und dem selektierten Namen einen Pfad herstellen, indem wir den Suchpfad mit diesem Dateinamen überschreiben und in den Rückgabestring fpn[] kopieren. Zum Schluß wird noch als Rückgabewert mitgeteilt, ob ABBRUCH (Rückgabe = 0)

Listing 3: Do__file in GFA-Basic

```
Nam$="abcdefgh.doc"
Pfad$="A:\"
Frm_key=1
While Frm_key=1
  Gosub Do_file(Pfad$,Nam$,"*.*",*Pfad$,*Nam$,*Gespfad$,*Taste)
  If Taste=0 Then
    Alert 1,"Taste ABBRUCH gedrückt !",1," OK | ABBRUCH ",Frm_key
  Else
    Alert 1,"Taste OK gedrückt !",1," OK | ABBRUCH ",Frm_key
  Endif
  If Frm_key=1 And Taste=1 Then
    Chdrive (Asc(Left$(Pfad$,1))-Asc("A")+1)
    Void Gemdos(59,L:Varptr(Pfad$))
    '     Chdir Pfad$
  Endif
Wend
'
Procedure Do_file(Fpath$,Finame$,Srch$,Fpath%,Finame%,Fpn%,Key%)
  '
  Local Tpath$,Tname$,I%
  Tpath$=Fpath$
  Tname$=Finame$
  If Finame$="" Then
    Fpath$=Chr$(Gemdos(25)+Asc("A"))+":"+Dir$(0)+"\"
  Endif
  Fpath$=Fpath$+Srch$
  Fileselect Fpath$,Finame$,Fpn$
  If Fpn$="" Then
    Fpath$=Tpath$
    Finame$=Tname$
    Key=0
  Else
    I=Len(Fpn$)
    While I>0 And Mid$(Fpn$,I,1)<>"\"
      Dec I
    Wend
    Fpath$=Left$(Fpn$,I)
    Finame$=Right$(Fpn$,Len(Fpn$)-I)
    Key=1
  Endif
  *Fpath%=Fpath$
  *Finame%=Finame$
  *Fpn%=Fpn$
  *Key%=Key
 Return
```

oder OK (Rückgabe = 1) angewählt worden war.

Wie Sie in Listing 2 und 3 sehen, haben wir diese recht nützliche Routine auch in Basic und auf mehrfachen Wunsch in PASCAL programmiert. Ich hoffe, daß die Erklärung anhand des C-Programms leicht auf die anderen Programme zu übertragen ist, zumal sie auch ausführlich dokumentiert sind. Wir haben versucht, die gleichen Bedingungen auch für BASIC zu schaffen, nur ist es allerdings zum 'Abspeichern' der neuen Pfade und Namen nötig, die Eingangsvariablen auch als Ausgangsvariablen zu definieren: Lästig, aber nicht anders zu machen, wenn man auf global gehaltene Variablen verzichten möchte. Noch ein Tip zur Handhabung: Die Übergabevariablen des Aufrufes sollten global definiert werden, da die Routine beim zweiten Aufrufen dann nicht den aktuellen Pfadnamen wieder heraussuchen muß, sondern den zuletzt benutzten verwenden kann. Jetzt werden Sie sich sicher fragen, ob denn der aktuelle und der mit der Fileselectorbox angewählte Pfad nicht der gleiche seien?! Aus der Fragestellung erkennt man schon, daß dies nicht der Fall ist. Nur dann, wenn sie explizit erreichen wollen, daß der von Ihnen ausgewählte Pfad zum aktuellen Pfad wird – auf den zum Beispiel auch ein Resource-Load zugreifen würde –, müssen Sie dies dem Betriebssystem mitteilen.

Auf anderen Wegen

Dies teilen Sie GEMDOS durch die beiden folgenden Befehle mit: Mit der Funktion Dsetdrv() übergeben Sie als Parameter die Diskettenstation, die zur aktuellen Diskettenstation gemacht werden soll. Den Pfad – ohne Laufwerksangabe – übergeben Sie dann der Funktion Dsetpath(). Damit dürften dem weiteren Programmieren in Bezug auf die Pfadverwaltung keine weiteren Probleme im Wege stehen.

Nicht nur eine Verschönerung

Viele unter Ihnen, die sich mit der Er-

stellung von Resourcen beschäftigt haben, werden sicher bemerkt haben, daß die Struktur des Icons recht aufwendig ist und daher leicht für Verwirrung sorgen kann. Deshalb wollen wir uns heute mit der **ICONBLK-Struktur** des Icons beschäftigen und dazu ein paar hilfreiche Routinen veröffentlichen; wir folgen damit einem Wunsch von vielen Lesern.

In Listing 4 befindet sich die erwähnte Struktur des Icons. Für diejenigen, die sich nicht mit 'C' auskennen, sei gesagt, daß LONG 4 Bytes und INT 2 Bytes bedeuten und vergleichbar mit einem RECORD in Pascal sind. Das Element ib__xchar würde man entsprechend an der 14. Stelle bezogen auf den Anfang der Struktur finden. Zunächst aber wollen wir uns wegen des besseren Verständnisses die Struktur eines IMAGES anschauen, da ein ICON ein IMAGE mit einigen zusätzlichen Features ist.

Tabelle 1:

DATA	MASK	Ergebnis
0	0	Hintergrund erscheint
0	1	Maskenfarbe
1	0	Iconfarbe
1	1	Iconfarbe

Listing 4: Die Iconblk-Struktur

```
typedef struct object
{
        int               ob_next;        /* -> nächstes Objekt */
        int               ob_head;        /* -> Kopf des Unterobjektes */
        int               ob_tail;        /* -> Ende des Unterobjektes */
        unsigned int ob_type;             /* Objekttyp*/
        unsigned int ob_flags;            /* Flags */
        unsigned int ob_state;            /* Status */
        char *ob_spec;                    /* spezielle Objektinfo */
        int               ob_x;           /* linke obere Ecke X- Koord. */
        int               ob_y;           /* linke obere Ecke Y- Koord. */
        int               ob_width;       /* Breite des Objektes */
        int               ob_height;      /* Höhe des Objektes */
} OBJECT;

typedef struct icon_block
{
        int *ib_pmask;                /* Adresse der Maskendaten */
        int *ib_pdata;                /* Adresse der Bilddaten */
        char *ib_ptext;               /* Adresse des Icontextes */
        int ib_char;                  /* Buchstabe incl. Farbe */
        int ib_xchar;                 /* X-Koordinate des Buchstabens */
        int ib_ychar;                 /* Y-Koordinate des Buchstabens */
        int ib_xicon;                 /* X-Koordinate des Icons */
        int ib_yicon;                 /* Y-Koordinate des Icons */
        int ib_wicon;                 /* Breite des Icons */
        int ib_hicon;                 /* Höhe des Icons */
        int ib_xtext;                 /* X-Koordinate des Textes */
        int ib_ytext;                 /* Y-koordinate des Textes */
        int ib_wtext;                 /* Breite der Textbox */
        int ib_htext;                 /* Höhe der Textbox */
} ICONBLK;

typedef struct bit_block      /* Struktur eines Images */
{
        int *bi_pdata;              /* Adresse der Bilddaten */
        int     bi_wb;              /* Breite des IMAGE in Bytes*/
        int     bi_hl;              /* Höhe in Punkten */
        int     bi_x;               /* X-Koordinate des Bildes*/
        int     bi_y;               /* Y-Koordinate des Bildes*/
        int     bi_color;           /* Farbe des IMAGE */
} BITBLK;
```

Zunächst befindet sich als erstes Element der IMAGE-Struktur ein Zeiger auf die Datenmenge des Bildes. In der Datenmenge entspricht einem gesetzten Bit einem Punkt auf dem Bildschirm, woraus automatisch folgt, daß nur zweifarbige Bilder (gesetzter und nichtgesetzter Punkt) möglich sind. An den Stellen, an denen kein Punkt gesetzt ist, wird später der Hintergrund durchscheinen! Als nächster Eintrag befindet sich in der Struktur die Breite des IMAGES in Bytes. Da sich die Daten in einem Wortfeld befinden, muß dieser Wert gerade sein. Als nächste Einträge folgen die Höhe des Bildes in Bildschirmpunkten, X- und Y-Koordinate desselben sowie ein Wort, das die Farbe des IMAGEs enthält.

Zwei Koordinaten für ein Objekt?

Zu den Koordinaten eines IMAGEs sollte noch ein wenig gesagt werden: Wie Sie sicher wissen, enthält jedes Objekt innerhalb des Baums einen Eintrag – so auch ein IMAGE oder ein ICON. Jedes Objekt hat allerdings auch eine X/Y-Koordinate sowie eine Höhe und eine Breite. Wie erwähnt, zeigt sich, daß in der IMAGE-Struktur (wie auch in der ICON-Struktur) eine zusätzliche X/Y-Koordinate vorhanden ist, woraus sich die Frage ergibt, warum eine zweifache Angabe der Koordinaten benötigt wird. Um Sinn und Zweck dieser doppelten Angabe zu verstehen, müssen wir uns daran erinnern, daß wir Objekte wie auch ICONS und IMAGES mit der Maus selektieren können. Da das AES überprüft, ob ein Objekt während eines Mausklicks 'getroffen' wurde, muß es auch den Umfang des Objektes kennen. Dieser Umfang wird mit der X/Y-Koordinate, Breite und Höhe angegeben, die sich in der OBJECT-Struktur des Objektes befinden. Die Koordinate des IMAGES (oder ICONS) in der speziellen Struktur gibt eine Verschiebung des Bildes relativ zur in der OBJECT-Struktur gegebenen Koordinate an. Praktisch heißt das, daß man die selektierbare Zone eines Bildes sehr groß machen kann und das Bild dann in die Mitte dieser Zone postiert. Ein Tip: Wollen Sie wissen, wie groß diese Zone ist, so stellen Sie in der OBJECT-Struktur des ICONS oder IMAGES 'OUTLINED' ein – dadurch wird dieser Bereich sichtbar. Noch ein kleiner Hinweis: Wie Sie gleich sehen werden, enthält ein ICON einen zusätzlichen Text. Achten Sie bitte darauf, daß sich der Text nicht außerhalb des selektierbaren Bereichs befindet, da ein Benutzer Ihres Programms den Text anklicken könnte und trotzdem nicht das Icon selektiert würde.

ICONheiten eines Icons

Womit wir beim Thema ICON wären: Was unterscheidet ein ICON von einem IMAGE? Der Unterschied liegt darin, daß man außer einer zusätzlichen Bildmanipulationsmöglichkeit einen BOXCHAR und einen STRING hinzugefügt hat. Am besten kann man sich dies an den Diskettenstationen des DESKTOPS verdeutlichen, wobei hier Buchstabe und Name des ICONs sinnvoll genutzt wurden.

Wie wir oben bei der Betrachtung der Bilddaten eines IMAGEs festgestellt haben, scheint bei einem nichtgesetzten Bit der Hintergrund durch. Um

dies gezielt zu verhindern, gibt es bei einem Icon eine zusätzliche Datenmenge, die Maske. Wird in dieser Maske an einer Stelle, an der in der Bildmenge kein Punkt gesetzt worden ist, ein Bit (Punkt) gesetzt, so scheint der Hintergrund an dieser Stelle nicht durch. Ist an dieser Stelle in der Maske kein Bit gesetzt, so scheint folgerichtig der Hintergrund wie gewohnt durch. In Tabelle 1 ist besonders deutlich zu sehen, daß die Maske nur bei einem nichtgesetzten Bit in den Bilddaten eine Auswirkung hat. Wird nun ein ICON selektiert, so erscheint das Bild der Maske in der Farbe des eigentlichen Bildes und umgekehrt. Ist zum Beispiel das Bild rot und die Maske grün, so ist nach dem Selektieren die Maske rot und das Bild grün.

Nachdem wir nun wissen, wie wir das ICON benutzen können, wollen wir uns die ICONBLK-Struktur genauer anschauen. Der erste Eintrag enthält die Adresse der MASKE, während im zweitem Eintrag die Adresse der Bilddaten steht. Der dritte Eintrag ib_ptext enthält einen Zeiger auf einen String, den Text des Icons. Der nächste Eintrag ib_char ist etwas komplizierter, da er nicht nur den Buchstaben des Icons enthält, sondern auch die Farben des Bildes und der Maske. Das Wort ist folgendermaßen aufgebaut:

Bild 2:

```
0000        0000        00000000
|            |                 |
  Vordergr.  Hintergr.  Buchstabe
```

"Farbwort des Icons: ib_char"

Zum Setzen des Wortes und Auseinanderziehen dieses Farbwortes habe ich die zwei Routinen make_color() und crack_color() in Listing 5 beigefügt, die diese Arbeit etwas erleichtern. Die folgenden Koordinaten des Buchstabens sind genauso relativ zu betrachten wie die nachfolgenden Koordinaten des ICONs und des Textes. Die Breite des Icons wird genauso wie bei dem IMAGE in Worten angegeben, wobei diese Zahl durch zwei teilbar sein muß. Auch die Höhe wird wie beim IMAGE in Bildschirmpunkten angegeben. Man sieht also, ein Icon bietet eine Menge an Möglichkeiten. Interessant sind noch die beiden letzten Einträge ib_wtext und ib_htext. Hiermit läßt sich sogar die Größe der Textbox einstellen, was leider oftmals nicht genutzt wird.

Listing 5

```
crack_color(cwort,parts)
int cwort;
CW *parts;
{
   parts->vorder= (cwort>>12)&0xf;
   parts->hinter= (cwort>> 8)&0xf;
   parts->buchstabe  = cwort&0xff;
}

int make_color(parts)
CW parts;
{

return(((parts.vorder)<<12)|((parts.hinter)<<8)|parts.buchstabe);
}
```

Nun möchte ich noch einige Ideen und Vorschläge zum Thema Icons einbringen. Icons sind nicht nur eine Verschönerung des Programms, sondern sollten auch zum besseren und schnelleren Verständnis häufiger als bisher eingesetzt werden. Nicht umsonst liegt auf dem DESKTOP ein Mülleimer in ICONFORM und nicht als reiner Text. Manchmal sagt ein kleines Bild mehr als viele Wörter und ist auch einprägsamer. Es stehen ihnen auch viele Möglichkeiten zu Verfügung: So können Sie zum Beispiel durch einfaches Verändern des Bilddatenzeigers und Neuzeichnen des Iconobjektes das Aussehen verändern. Wie wäre zum Beispiel ein Mülleimer in einem Ihrer Programme, der, wenn er gefüllt ist, auf einmal einen geschlossen Deckel hat. Übrigens brauchen Sie natürlich nicht für jedes Icon eine neue Bilddatenmenge. Stellen Sie sich vor, was das für das DESKTOP zur Folge hätte, denn teilweise sind dreißig und mehr Icons zu sehen. Sie lassen einfach mehrere Icons auf eine Bilddatenmenge zeigen.

Zum Schluß möchte ich Sie noch auf etwas hinweisen. Wenn Sie mit dem RCS (Resource Construction Set) von Atari arbeiten, werden Sie feststellen, daß Sie zwar Icons benutzen, diese aber nicht im RCS erstellen können. Dies geschieht mit einem sogenannten Iconeditor der verschiedensten Firmen, der es ermöglicht, eine bestimmte Datenmenge abzuspeichern, die Sie dann mit dem RCS laden können. Dabei verfahren Sie folgendermaßen: Legen Sie ein Icon in Ihre Dialogbox und selektieren Sie es. Dann können Sie unter dem Menü OPTIONS die Option LOAD anwählen. Darauf können Sie die Bilddaten und Maskendaten nacheinander laden.

Damit möchte ich mich für diesen Monat verabschieden, allerdings nicht, ohne Sie einmal mehr darauf hinzuweisen, daß Sie mir Ihre Probleme und Nöte bei der täglichen Programmierarbeit mit dem ST gerne schicken können. Übrigens, wenn Sie mitgeteilt bekommen möchten, wann ich mich Ihres Problems annehme oder ob ich es schon in anderen Ausgaben der ST-Computer getan habe, so bitte ich Sie, einen frankierten und adressierten Rückumschlag beizulegen. Und bitte, vergessen Sie bei Ihrer Anschrift die Telefonnummer nicht, denn manche Probleme könnten am Telefon sehr schnell geklärt werden. Viel Spaß beim Programmieren und bis zum nächsten Monat!

(SH)

Blitter TOS

Das neue TOS im Atari ST

Allgemeines

Gleichzeitig mit der Auslieferung des MEGA ST hat Atari auch eine neue Version des Betriebssystems vorgestellt. Die sechs Roms aus dem MEGA ST enthalten in 192 K-Byte das komplette TOS. Um eine weite Verbreitung des neuen TOS zu ermöglichen, ist es auch im alten ST-Modell einsetzbar, da es automatisch erkennt, um welchen Rechner es sich handelt. Wir wollen näher darauf eingehen, in welchen Teilen das Betriebssystem Änderungen erfahren hat und welche Teile neu hinzugekommen sind.

Das TOS hat die Versionsnummer 1.2 mit Datum vom 22.4.1987. In den ausgelieferten Geräten in der Bundesrepublik befindet sich die deutsche Version, die sich durch einen modifizierten Tastaturtreiber und deutsche Texte von der amerikanischen Version unterscheidet. Es ist (fast) kompatibel zum alten TOS, d. h., alle sauber geschriebenen Programme sollten auch laufen.

Auf den ersten Blick gibt es folgende Neuerungen: Verbesserte Benutzerfreundlichkeit des Desktop, Beschleunigung der Grafik durch den Blitter, Verwaltung der residenten Uhr, Erweiterung von Betriebssystem-Funktionen, Beseitigung einiger Fehler, Optimierung in Bezug auf Programmlänge und Geschwindigkeit.

Dieser Artikel stützt sich im wesentlichen auf eigene Untersuchungen und nicht auf das von Atari herausgegebene Dokumentationsblättchen, da es Atari scheinbar nicht möglich ist, vollständige Informationen über das Betriebssystem zur Verfügung zu stellen.

Wir werden hauptsächlich über das BIOS und das GEMDOS berichten. Die Erweiterungen und Ergänzungen des BIOS (ausgenommen der Bildschirm-Treiber) sind (hoffentlich) vollständig beschrieben. Am GEMDOS wurden außer den bei „Uhrzeit und Datum“ und „Optimierungen“ beschriebenen Kleinigkeiten keine Änderungen vorgenommen. Am GEM und Desktop wurde erheblich mehr geändert, doch liegen hier erst wenige Erkenntnisse über die internen Veränderungen vor. Die detaillierteren Informationen richten sich hauptsächlich an solche Leser, die schon über Programmierkenntnisse des TOS verfügen. Zum vollen Verständnis der Erklärungen zur Speicher- und Programmverwaltung sei auf den Artikel „TOS intern“ im neuen ST-Sonderheft verwiesen.

Speicherbelegung

Das neue TOS reserviert sich vom RAM zusätzliche 10496 Byte für den eigenen Bedarf, wovon allein 10000 Byte für die Vergrößerung des internen GEMDOS-Speichers gebraucht werden. AES und Desktop benötigen 256 Bytes zusätzlich, der Rest wird für einige zusätzliche BIOS-Variablen für die neuen Fähigkeiten des TOS gebraucht. Im einzelnen zeigt die Systemvariable savptr ($4A2) nach dem Reset weiterhin auf $93A, die BIOS- und GEMDOS-Variablen gehen bis $88A2 (dies wurde auf $8900 aufgerundet) und die AES/Desktop-Variablen reichen bis $CA00.

Die Aufteilung der 192 kB ROM hat sich ebenfalls geändert. Um im ROM Platz für die neuen Blitter- und Uhrzeit-Routinen zu bekommen, mußte irgendwo gespart werden. Die naheliegenden Möglichkeiten wie Verwendung der „Short-Adressierung“ beim Zugriff auf die Systemvariablen und die Peripherie-Bausteine, das Streichen überflüssiger, d. h. nicht benutzter Routinen, Neuprogrammierung der verschwenderischen Hardcopy-Routine usw. wurden nicht genutzt. Einige BIOS-Routinen wurden ein klein wenig umgeschrieben, aber das macht nur wenige Bytes Ersparnis aus. BIOS und GEMDOS sind durch die kleinen Ergänzungen um einige Bytes länger geworden.

Doch woher stammt der benötigte Platz? Zum einen wurden der BIOS-Bildschirm-Treiber, Line A und das VDI gründlich überarbeitet. Dabei wurde teilweise optimaler programmiert, viele Routinen sind kaum wiederzuerkennen. Außerdem wird jetzt beim Zugriff auf die BIOS/Line A/VDI-Variablen platzsparender verfahren (Register-indirekt- statt Absolut-Adressierung), dies spart pro Zugriff immerhin 2 Bytes. Da in diesem Teil die sehr langen Blitter-Routinen untergebracht sind, sind Line A und VDI trotz der Optimierungsmaßnahmen noch 4,4 kB länger geworden. Erstaunlicherweise wurden im AES/Desktop ca. 3,6 kB eingespart. Wie das erreicht wurde, ist noch unbekannt. Hier wurden aber offensichtlich ebenfalls größere Änderungen vorgenommen, wie man an den sichtbaren Auswirkungen sieht.

Optimierungen

Neben der Verkürzung des TOS wurden auch einige zeitliche Optimierungen vorgenommen:

Das Löschen des Speichers beim Einschalten des Rechners wurde um 19 % beschleunigt, bei 4 MB dauert dies aber immer noch ca. 1,7 Sekunden.

Beim Laden eines Programms wird der restliche freie Speicher gelöscht. Dies war sehr langsam, aber die dadurch entstandene Pause von 1,9 s (bei einem „leeren“ ST) fiel wegen der Nachlaufzeit der Floppy nicht so auf. Bei 4 MB-Rechnern traten allerdings unangenehm lange Pausen beim Starten eines Programms auf. Das TOS hat jetzt eine sehr schnelle, 6 mal schnellere Speicher-Löschroutine, so daß die Zeit für das Speicherlöschen bei einem „leeren“ 4 MB-ST „nur“ 1,4 s dauert. Bei künftigen TOS-Versionen und „Super-STs“ mit noch mehr RAM sollte sich Atari allerdings überlegen, ob das Löschen des Speichers nicht ganz weggelassen werden sollte, da es eigentlich überflüssig ist (bei sauber geschriebenen Programmen); es reicht nämlich, nur das BSS-Segment des geladenen Programms zu löschen.

Das Bildschirm-Löschen des BIOS und VDI wurde mit der gleichen Routine beschleunigt (11,3 ms statt 18 ms).

Durch die Überarbeitung des Bildschirm-Treibers und des GEM sind die meisten Bildschirmausgaben schneller geworden.

Beseitigung von Fehlern

Mit der Erstellung einer neuen TOS-Version hatte Atari eine gute Gelegenheit, Fehler zu beseitigen. Dies wurde

leider nur teilweise genutzt, es wurden nämlich nur einige Fehler im BIOS (und wohl auch im GEM) behoben. An das von Digital Research stammende GEMDOS haben sich die Programmierer von Atari wohl nicht herangetraut, denn dort wurde kein einziger Fehler beseitigt, obwohl es mehr als genug davon gibt. Auch im BIOS gibt es noch Fehler, die nicht beseitigt wurden.

Zum GEMDOS gibt es nur zu sagen, daß die Größe des internen GEMDOS-Speichers von 3000 auf 8000 Words erhöht wurde. Dies hat zur Folge, daß die „Fehler" in der Speicherverwaltung und im Zusammenhang mit vielen Ordnern, die ja eigentlich gar keine Programmierfehler sind, sondern „nur" aus Speicherplatzmangel entstehen, behoben sind.

Doch nun zu den Änderungen im BIOS:

In der Reset-Routine wurde die fehlerhafte Abfrage des PAL/NTSC-Flags aus dem TOS-Header korrigiert. Bei Farbdarstellung ist der Default-Wert für die Bildwiederholfrequenz bei der deutschen TOS-Version jetzt also 50 Hz.

Der Bildschirm wird jetzt zu Beginn des Speicherlöschens beim Einschalt-Reset gelöscht und nicht erst danach.

Die nicht benutzten Auto-Interrupt-Vektoren 1,3,5 und 7 waren bisher undefiniert. Nun werden solche Interrupts, die eigentlich gar nicht auftreten können, ignoriert.

Wenn beim Lesen oder Schreiben ein Fehler auftritt, wird die Operation mehrmals wiederholt, bevor mit einer Fehlermeldung abgebrochen wird. Wenn eine ganze Gruppe von Sektoren hintereinander geschrieben oder gelesen wird, war bisher die Anzahl der Versuche auf die ganze Gruppe bezogen. Konnte also der erste Sektor z. B. erst nach dem dritten Versuch gelesen werden, so wurde beim zweiten Sektor schon nach dem ersten Versuch abgebrochen. Nun gibt es bei jedem Sektor drei Versuche. Die Korrektur dieses Fehlers hat weiterhin zur Folge, daß das BIOS den Floppy-Controller nun alle Sektoren einzeln lesen läßt und den „multiple read"-Befehl nicht mehr verwendet.

Beim Lesen eines Sektors von Diskette werden jetzt zusätzlich vom Floppy-

```
;Änderungen des ATARI ST-ROMS vom 22.4.1987 (US-Version)
;in BIOS (außer Bildschirmtreiber) und GEMDOS
;
;ROM-Header
FC0000 BRA      $FC0030
FC0002 dc.b     1,2                   ;Version 1.2
FC0004 dc.l     $FC0030               ;Reset-Adresse
FC0008 dc.l     $FC0000               ;Beginn TOS
FC000C dc.l     $008900               ;Beginn freies RAM
FC0010 dc.l     $FC0030               ;Default-Shell
FC0014 dc.l     $FEFFF4               ;Adresse GEM-MAGIC
FC0018 dc.l     $04221987             ;Default-Systemdatum
FC001C dc.w     0                     ;NTSC-Flag
FC001E dc.w     $0E96                 ;Default-Systemdatum im GEMDOS-Format
FC0020 dc.l     $007e9c               ;Adresse GEMDOS-Variable 'mifl'
FC0024 dc.l     $000E61               ;Adresse BIOS-Tastaturvariablen
FC0028 dc.l     $0087CE               ;Adresse GEMDOS-Variable 'act_pd'
FC002C dc.l     0                     ;reserviert ?
;
FC0030 ...                            ;Beginn Reset-Routine: unverändert
FC009E BTST     #0,$FC001D(PC)        ;PAL/NTSC-Flag richtig abgefragt
FC0154 MOVE.B   D6,$FF8001            ;unverändert
FC015A LEA      $008000,A7            ;SP setzen, wozu ?
FC0160 ...                            ;unverändert
FC01A2 MOVE.L   A4,$000008            ;unverändert
FC01A8 MOVE.L   D5,D0                 ;Speicherobergrenze
FC01AA SUB.L    #$00008000,D0         ;- 32 KB
FC01B0 LSR.W    #8,D0                 ;als Bildschirm-Adresse setzen
FC01B2 MOVE.B   D0,$FF8203
FC01B8 SWAP     D0
FC01BA MOVE.B   D0,$FF8201
FC01C0 MOVE.L   D5,A0                 ;Speicher löschen (19% schneller)
FC01C2 MOVE.L   #$00000400,D4         ;0,43 statt 0,52 Sek. pro MB
FC01C8 MOVEQ    #$00,D0
FC01CA MOVEQ    #$00,D1
FC01CC MOVEQ    #$00,D2
FC01CE MOVEQ    #$00,D3
FC01D0 MOVEM.L  D0-D3,-(A0)
FC01D4 MOVEM.L  D0-D3,-(A0)
FC01D8 MOVEM.L  D0-D3,-(A0)
FC01DC MOVEM.L  D0-D3,-(A0)
FC01E0 CMPA.L   D4,A0
FC01E2 BNE      $FC01D0
FC01E4 SUBA.L   A5,A5                 ;unverändert
FC01E6 MOVE.B   D6,$0424(A5)
FC01EA MOVE.L   D5,$042E(A5)
FC01EE MOVE.L   #$752019F3,$0420(A5)
FC01F6 MOVE.L   #$237698AA,$043A(A5)
FC01FE MOVE.L   #$5555AAAA,$051A(A5)  ;neues memvalid, wozu ?
FC0206 ...                            ;unverändert
FC02FA MOVE.L   #$00FC0670,$046E(A5)  ;unverändert: swv_vec setzen
FC0302 CLR.L    $0004C2               ;alle Laufwerke abmelden
FC0308 BSR      $FC0EBA               ;ROM-Header im RAM anlegen
FC030C ...                            ;unverändert
FC0340 MOVE.L   A3,$000014            ;unverändert: Division by Zero-Trap
FC0346 MOVEQ    #$06,D0               ;Auto-Vektoren 1...7 auf RTE setzen
FC0348 LEA      $0064(A5),A1
FC034C MOVE.L   #$00FC07CE,(A1)+
FC0352 DBF      D0,$FC034C
FC0356 ...                            ;unverändert
                                      ;nach vblqueue-Initialisierung
FC03A4 LEA      $FC09AE,A0            ;BIOS-Device-Vektoren setzen
FC03AA MOVE.W   #$051E,A1
FC03AE MOVEQ    #$1F,D0
FC03B0 MOVE.L   (A0)+,(A1)+
FC03B2 DBF      D0,$FC03B0
```

Controller gemeldete CRC-Fehler erkannt, sie wurden bisher ignoriert.

Wenn bei einem Diskettenzugriff ein CRC-Fehler erkannt wurde, lieferte das BIOS die Default-Fehlermeldung, das sind –10 (Schreibfehler) oder –11 (Lesefehler), anstelle der eigentlich vorgesehenen Fehlermeldung –4 (CRC-Fehler). Umgekehrt lieferte es einen CRC-Fehler, wenn der Default-Fehler auftreten sollte. Dieser Fehler ist nun behoben.

Die RS232-Routinen waren wohl so fehlerhaft, daß sich kleine Korrekturen nicht mehr lohnten. Sie wurde nämlich komplett neu programmiert. Ob die RS232-Schnittstelle jetzt richtig funktioniert, konnte noch nicht eindeutig festgestellt werden.

Beim Reset wird der Tastaturprozessor nur noch mit der Byte-Folge $80, $01 statt $80, $01, $12, $1A initialisiert. Außerdem wird danach eine Pause von ca. 0,18 s eingelegt.

Beim Senden eines Bytes an den Tastaturprozessor wird ca. 5,2 ms lang gewartet, nachdem er als empfangsbereit erkannt wurde.

Die Systemvariable drvbits ($4C2) wird nun in der Reset-Routine gelöscht. Damit werden alle zusätzlichen Laufwerke wie RAM-Disks usw. beim Reset ordnungsgemäß abgemeldet.

Neue TOS-Funktionen

Das BIOS wurde um eine Funktion für den Blitter erweitert. Der Funktionsumfang von GEMDOS, VDI und AES wurde nicht vergrößert, insbesondere die im alten TOS nicht implementierten VDI-Funktionen des PC-DOS-GEM fehlen weiterhin. Allerdings ist es nicht auszuschließen, daß einzelne Funktionen erweitert wurden, ähnlich wie die Formatier-Funktion des BIOS (s. u.).

Die von der GEMDOS-Funktion Sversion gelieferte GEMDOS-Versionsnummer ist weiterhin $1300.

Der Blitter

Der Blitter ist bekanntlich ein in die MEGA-STs eingebauter neuer Custom-Chip, der die Grafik erheblich beschleunigt. Die Treiber-Routinen sind in den Grafik-Kern des ST (Line A und BIOS-Bildschirm-Treiber) integriert, so daß alle Programme, die ihre Grafik durch die vorgesehenen TOS-Funktionen (die letztlich alle auf den Grafik-Kern zurückgreifen) auf den Bildschirm bringen, automatisch in den „Genuß" des Blitters kommen. Programme, die die Grafik ohne Unterstützung durch das TOS erzeugen, werden also durch den Blitter nicht beschleunigt.

```
FC03B6 BSR      $FC2408                   ;MFP init. (außer IKBD-Reset)
FC03BA MOVE.L   #$00FC053A,-(A7)          ;IKBD-Reset: $80,$01 senden
FC03C0 MOVE.W   #$0001,-(A7)              ;Mouse+Joystick disable fehlt
FC03C4 JSR      $FC2212
FC03CA ADDQ.L   #6,A7
FC03CC MOVE.L   #$00007FFF,D0             ;ca. 0,18 Sek. warten
FC03D2 BSR      $FC0556                   ;-> RTS
FC03D6 DBF      D0,$FC03D2
FC03DA MOVEQ    #$02,D0                   ;unverändert: Bildschirmauflösung setzen
FC03DC ...
FC0418 BSR      $FC0F1A                   ;Test ob Blitter vorhanden
FC041C JSR      $FCA99E                   ;Blitter-Vektoren und -Status init.
FC0422 JSR      $FCA914                   ;unverändert: Bildschirmausgabe init.
FC0428 ...
FC045C BSR      $FC4A9E                   ;unverändert: GEMDOS init.
FC0460 MOVE.W   $FC001E,$008840           ;Systemzeit init.
FC046A BSR      $FC4BE8                   ;Systemzeit von Uhrenchip übernehmen
FC046E BSR      $FC053C                   ;unverändert: Floppy-Boot
FC0472 BSR      $FC0558                   ;             DMA-Boot
FC0476 BSR      $FC0E56                   ;             reset-residente Programme
FC047A TST.W    $000482
FC0480 BEQ      $FC04A0
                                          ;cmdload: Cursor einschalten fehlt
FC0482 BSR      $FC0C4A                   ;Auto-Ordner ausführen
FC0486 MOVE.L   #$00FC0000,$0004F2        ;sysbase wieder auf ROM-Header
FC0490 PEA      $FC0537(PC)               ;unverändert
FC0494 PEA      $FC0537(PC)
FC0498 PEA      $FC0524(PC)
FC049C CLR.W    -(A7)
FC049E BRA      $FC0508
FC04A0 BSR      $FC0C4A
FC04A4 MOVE.L   #$00FC0000,$0004F2        ;sysbase wieder auf ROM-Header
FC04AE ...                                ;unverändert
                                          ;Strings bis  'GEM.PRG',0,0,0
FC053A dc.b     $80,$01                   ;String für IKBD-Reset
FC053C ...                                ;unverändert
;
;
;neue Routinen
;
;neue BIOS-Routine Gettime
FC0E96 LEA      $FC4C44,A3                ;für Uhrenchip
FC0E9C LEA      $FC1F52,A4                ;für IKBD (altes Gettime)
FC0EA2 BRA      $FC0EB0
;neue XBIOS-Routine Settime
FC0EA4 LEA      $FC4D02,A3                ;für Uhrenchip
FC0EAA LEA      $FC1F6C,A4                ;für IKBD (altes Settime)
FC0EB0 BSR      $FC4C0C                   ;Uhrenchip vorhanden ?
FC0EB4 BCC      $FC0EB8                   ;-> ja
FC0EB6 MOVE.L   A4,A3
FC0EB8 JMP      (A3)
;
;ROM-Header im RAM anlegen
FC0EBA LEA      $FC0000(PC),A0            ;48 Byte kopieren
FC0EBE LEA      $000940,A1
FC0EC4 MOVEQ    #$2F,D0
FC0EC6 MOVE.B   $00(A0,D0.W),$00(A1,D0.W)
FC0ECC DBF      D0,$FC0EC6
FC0ED0 MOVE.W   $FC0EEE(PC),$FFFA(A1)     ;JMP-Befehl vor Header-Kopie
FC0ED6 MOVE.L   $0004(A1),$FFFC(A1)       ;nach Reset-Adresse auf Header
FC0EDC MOVE.W   $FC0EF4(PC),(A1)          ;Branch auf JMP-Befehl an Header-Anfang
FC0EE0 MOVE.W   $001E(A1),$001C(A1)       ;Default-Datum nach PAL-Flag
FC0EE6 MOVE.L   A1,$0004F2                ;sysbase auf Header-Kopie setzen
FC0EEC RTS
FC0EEE JMP      $000000
FC0EF4 BRA      $FC0EEE
```

Der Blitter belegt die Adressen ab $FF8A00. Da bei den alten STs ein Zugriff auf diesen Bereich zu einem Bus Error führt, kann das TOS erkennen, ob ein Blitter eingebaut ist, indem es Blitter-Register ausliest und sich merkt, ob ein Bus Error auftritt.

Intern hat das TOS eine Liste mit 10 Sprungvektoren für 10 elementare Grafik-Funktionen wie z. B. „ein Zeichen auf dem Bildschirm ausgeben". Jede BIOS-Bildschirm-Treiber- oder Line A-Funktion führt nun indirekte Sprünge über diese 10 Sprungvektoren aus, wann immer eine elementare Grafik-Funktion gebraucht wird. Die 10 Funktionen existieren je zweimal, einmal mit und einmal ohne Verwendung des Blitters. So kann durch einfaches Auswechseln der Sprungliste der Blitter aktiviert und deaktiviert werden.

Um den Grafik-Kern von Programmen aus „umzuschalten", wurde die XBIOS-Funktion 64 namens Blitmode implementiert. Ihr wird ein Word-Parameter übergeben, der folgende Werte annehmen darf:

Parameter = −1:
Die Funktion gibt den Blitter-Status zurück. Ein gesetztes Bit 1 gibt an, daß überhaupt ein Blitter vorhanden ist, es sich also um einen MEGA-ST handelt. Ein gesetztes Bit 0 zeigt an, daß der Blitter auch tatsächlich aktiv ist, d. h. vom Grafik-Kern des TOS benutzt wird. Bei den alten STs sind Bit 0 und 1 dementsprechend immer Null. Die anderen Bits haben keine Bedeutung.

Parameter = 0:
Der Blitter wird ab jetzt nicht mehr vom TOS benutzt. Außerdem gibt die Funktion den Blitter-Status vor dem Abschalten des Blitters zurück.

Parameter = 1:
Der Blitter wird aktiviert. Falls gar keiner vorhanden ist, ändert sich natürlich nichts. Hier wird ebenfalls der Blitter-Status vor der Aktivierung zurückgeliefert.

Das BIOS schaltet den Blitter übrigens beim Reset in jedem Fall ab.

Es bleibt zu hoffen, daß diese neue Funktion möglichst bald in die Bindings der Compiler aufgenommen wird.

Uhrzeit und Datum

Bekanntlich sind die neuen Mega-Ataris mit einer residenten Uhr ausgerüstet. Dabei handelt es sich aber nicht um die Uhr des Tastaturprozessors,

sondern Atari hat einen eigenen Uhrenchip eingebaut.

Er liegt an den ungeraden Adressen ab $FFFC21. Bei den alten STs ergibt ein Zugriff keinen Bus Error, sondern $FF. Das TOS erkennt das Vorhandensein des Uhrenchips, indem es Register auf bestimmte Werte überprüft.

Diese Hardware-Uhr muß nun mit der softwaremäßigen, interruptgesteuerten Systemuhr des GEMDOS, die für die in Diskettenverzeichnissen eingetragene und von Programmen angezeigte Zeit zuständig ist, ständig übereinstimmen.

Beim Reset wird die Systemuhr nach der Hardware-Uhr gestellt, falls sie vorhanden ist.

Wenn die Systemuhr mit den GEMDOS-Funktionen Tsettime oder Tsetdate gestellt wird, wird die Hardware-Uhr ebenfalls gestellt. Bei den alten STs wird dabei die Uhr des Tastaturprozessors auf die neue Zeit gesetzt. Dies war beim alten TOS nicht der Fall, dort wurde die Tastaturprozessor-Uhr von GEMDOS überhaupt nicht benutzt. Diese kleine Inkompatibilität wird allerdings bei bestehenden Programmen kaum etwas ausmachen.

Tgettime und Tgetdate lesen in jedem Fall die Systemuhr aus. Die XBIOS-Funktion #22 (Settime) bzw. #23 (Gettime) setzt bzw. liest bei den MEGA-STs nun die neue Hardware-Uhr, bei den alten STs wird wie beim alten TOS die Tastaturprozessor-Uhr angesprochen. Bei den neuen STs kann die Tastaturprozessor-Uhr also gar nicht mehr benutzt werden, außer natürlich mit direkten Kommandos an den Tastaturprozessor. Bei den alten STs hat Settime beim alten und neuen TOS keinerlei Einfluß auf die Systemuhr, bei den neuen STs wird bei jedem Programmende (Pterm) die Hardware-Uhr in die Systemuhr übertragen. Programme, die die über das XBIOS angesprochene Uhr bisher für eigene Zwecke gebraucht haben, verstellen also bei den neuen STs die Systemuhr!

```
;XBIOS-Routine #64 (Blitter an-/ausschalten)
;Parameter: neuer Blitter-Status oder -1 für Abfrage
FC0EF6 BSR     $FC0F1A                  ;Test ob Blitter vorhanden
FC0EF8 MOVE.W  D0,D4
FC0EFA MOVE.W  D0,D5
FC0EFC LSR.W   #1,D5                    ;Blitter-vorhanden-Flag nach Bit 0
FC0EFE OR.W    #$FFFE,D5                ;Maske für 'Blitter an'
FC0F02 BSR     $FC4E06                  ;Blitter-Status holen
FC0F06 MOVE.W  D0,D3                    ;und merken
FC0F08 MOVE.W  $0004(A7),D0             ;Parameter
FC0F0C BMI     $FC0F16                  ;-> Blitter-Status zurückgeben
FC0F0E AND.W   D5,D0                    ;Blitter an nur wenn Blitter vorhanden
FC0F10 OR.W    D4,D0                    ;Blitter-vorhanden-Flag dazu
FC0F12 BSR     $FC4DDE                  ;Blitter-Vektoren setzen
FC0F16 MOVE.W  D3,D0                    ;Status zurückgeben
FC0F18 RTS
;
;Test ob Blitter vorhanden
FC0F1A MOVE.W  SR,D1
FC0F1C MOVE.W  #$0000,D0                ;Status: kein Blitter
FC0F20 SUBA.L  A0,A0
FC0F22 MOVE.L  A7,A2
FC0F24 ORI.W   #$0700,SR                ;IPL 7
FC0F28 MOVE.L  $0008(A0),A1
FC0F2C MOVE.L  #$00FC0F3A,$0008(A0)     ;Bus Error abfangen
FC0F34 TST.W   $8A00(A0)                ;versuchter Zugriff auf Blitter
FC0F38 MOVEQ   #$02,D0                  ;ok: Status: Blitter vorhanden
FC0F3A MOVE.L  A1,$0008(A0)             ;Zustand vor Bus Error wieder herstellen
FC0F3E MOVE.W  D1,SR
FC0F40 MOVE.L  A2,A7
FC0F42 RTS

geänderte BIOS-Routinen
FC0688  Test auf gültige Speicherkonfiguration
         zusätzlich memvalid #3 bei $51A abfragen
FC0984  BIOS-Device-Funktionen
         Sprung über neue RAM-Vektoren
FC1782  floprd
         Sektoren einzeln von FDC lesen (kein multiple read mehr)
         3 Versuche bei jedem Sektor statt für ganze Sektorgruppe
         zusätzlich führt Bit2 des FDC-SR zu Fehler (wozu das?)
FC1858  flopwr
         3 Versuche bei jedem Sektor statt für ganze Sektorgruppe
FC1916  flopfmt + fmtrack
         Bei interleave = -1 ist 'filler' Zeiger auf Liste mit Sektornummern
         (beliebige Sektornummern möglich)
FC1C48  floplock
         Berechnung von 'edma' aus 'cdma' und 'ccount' fehlt
FC1F08  Diskflags
         fehlen
FC1FDA  Binär<->BCD-Konvertierungen für IKBD-Zeit-Umwandlungen
         kürzer programmmiert
FC2150  RS232-Ein-/Ausgabe
         komplett neu programmiert
FC21EE  ikbdwc (Byte an IKBD senden)
         Pause von ca. 5,2 ms zwischen 'ACIA bereit' und 'Byte senden'
FC2408  initmfp
         CTS auf High setzen
         IKBD-Initialisierung fehlt (s. Reset)
FC26E6  bselect
         kürzer programmiert
FC26F6  RS232
         komplett neu programmiert
FC290E  rsconf
         geändert
```

Erweiterte Disk-Formatier-Funktion

Die XBIOS-Funktion 10 (Spur formatieren) wurde erweitert. Die folgende neue Aufrufmöglichkeit ist:

Flopfmt (puffer, sectlist, dev, spt, track, seite, −1, magic, virgin)

Die Parameter puffer, dev, spt, track, seite, magic, virgin haben die übliche Bedeutung. Der „Interleaving-Faktor" −1 führt dazu, daß 'sectlist' (nomalerweise unbenutzt) als Zeiger auf eine Tabelle mit soviel Sektornummern (words), wie 'spt' angibt, interpretiert wird. Die Spur wird aus den in dieser Liste angegebenen Sektoren in der spezifizierten Reihenfolge zusammengesetzt. Man kann also die Nummern und Reihenfolge der Sektoren vollkommen frei wählen. Wenn die Diskette später mit den normalen BIOS-Funktionen gelesen werden soll, muß man natürlich darauf achten, daß jede Sektornummer von 1 bis spt genau einmal vorkommt.

Erweiterter TOS-Header

Der TOS-Header, der gleich am Anfang des TOS liegt, enthält 4 zusätzliche Adressen, so daß der Rest des TOS um 16 Byte verschoben ist. Er sieht nun folgendermaßen aus:

```
FC0000 BRA  $FC0030    ;Sprung auf Beginn Reset-Routine
FC0002 dc.b 1,2        ;Version 1.2
FC0004 dc.1 $FC0030    ;Reset-Adresse
FC0008 dc.1 $FC0000    ;Beginn des TOS
FC000C dc.1 $008900    ;Beginn freies RAM
FC0010 dc.1 $FC0030    ;Default-Shell (falls kein GEM da)
FC0014 dc.1 $FEFFF4    ;Adresse des GEM-Magic
FC0018 dc.1 $04221987  ;Default-Systemdatum 22.4.1987
FC001C dc.w 0          ;NTSC-Flag (beim amerikanischen TOS)
FC001E dc.w $0E96      ;Default-Systemdatum im GEMDOS-Format
FC0020 dc.1 $007E9C    ;Zeiger auf den internen Speicher des GEMDOS'mifl
```

Bei Version 1.0 sind Schwierigkeiten mit dem internen GEMDOS-Speicher aufgetreten, da nur 6000 Bytes dafür reserviert waren. Deshalb ist das Problem mit Festplatten aufgetreten, wenn diese sehr viele Ordner enthielten, und auch die Funktion Malloc konnte nur sehr sparsam benutzt werden, da jedes Anlegen eines Speicherbereichs einen Eintrag in der internen GEMDOS-Speicherliste benötigt, die schnell überfüllt war. Beim TOS 1.2 besitzt der GEMDOS-Speicher nun eine Größe von 16000 Bytes, was für normale Anwendungen ausreichend sein sollte.

'mifl' dient der Erweiterung des internen GEMDOS-Speicherbereichs, falls 8000 Words immer noch nicht genug sein sollten. Interessanter wäre die Adresse des Memory Parameter Block gewesen, da dann Programme die legale Möglichkeit gehabt hätten, die Speicherverwaltung zu manipulieren.

FC0024 dc.1 $000E61 ;Adresse der BIOS-Variablen 'kb__shift'

'kb__shift' (Byte) enthält den Status der Sondertasten der Tastatur, wie ihn auch die BIOS-Funktion 11 (Kbshift) liefert. Damit ist es möglich, eigene Tastatur-Interrupt-Routinen zu schreiben, die den Tasten-Status benötigen und ändern wollen. Direkt im Anschluß an 'kb__shift' liegen noch weitere Tastatur-Variablen, allerdings ist nicht bekannt, ob deren relative Adressen zu 'kb__shift' von Atari für spätere TOS-Versionen garantiert sind. Deswegen werden sie hier nicht angegeben.

FC0028 dc.1 $0087CE ;Adresse der GEMDOS-Variablen 'act__pd'

'act__pd' ist der Zeiger auf den Prozeß-Deskriptor des aktiven Prozesses. Hiermit bieten sich neue Möglichkeiten, Multi-Tasking zu realisieren, da GEMDOS dies prinzipiell ermöglicht. Beim Umschalten zwischen zwei Programmen müßte 'act__pd' immer auf den aktiven Prozeß gesetzt werden, dann würden sich alle GEMDOS-Kommandos für Speicherverwaltung, Diskettenoperationen usw. automatisch immer auf den richtigen Prozeß beziehen! Verschiedene Programme können gleichzeitig Speicherbereiche belegen, eigene Dateipfade und Dateien haben usw. Nur die Zeichen-orientierte Peripherie wäre allen gemeinsam. Dann müßte man nur noch GEM die gleichzeitige Verwaltung mehrerer Haupt-Applikationen beibringen...

FC002C dc.1 0 ;reserviert?

Beim Reset wird eine Kopie des TOS-Headers im RAM angelegt. Diese Kopie wird so modifiziert, daß beim Sprung auf ihren Beginn ein Reset im ROM ausgeführt wird, genau wie wenn man den TOS-Header im ROM direkt anspringt. Dies wird erreicht, indem der BRA-Befehl des Headers auf einen JMP-nach-Reset-Befehl (unmittelbar vor dem Header) führt. Weiterhin wird das PAL/NTSC-Flag (relative Adresse $1C) mit dem Default-Systemdatum (relative Adresse $1E) überschrieben, es fragt sich nur, wozu. Außerdem sind nach dem Header noch 16 Bytes freigelassen worden.

Die BIOS-Variablen sysbase ($4F2) wird ebenfalls beim Reset auf diese TOS-Header-Kopie gesetzt. Damit haben Module im EPROM-Port die Möglichkeit, den TOS-Header zu verändern und anschließend in die normale Reset-Routine zurückzuspringen. In der Reset-Routine werden nämlich verschiedene Daten aus dem Header benutzt. Es ist allerdings zwecklos, die drei neuen Variablen-Adressen zu manipulieren, da BIOS und GEMDOS direkt auf die durch sie bezeichneten Variablen zugreifen. Erst nach dem Abarbeiten des AUTO-Ordners und vor dem Start des GEM oder eines COMMAND.PRG mittels der cmdload-Option des Bootsektors, wird sysbase wieder auf den TOS-Header im ROM gesetzt. Somit haben also auch Programme im AUTO-Ordner die Möglichkeit, den TOS-Header zu manipulieren, es fragt sich nur, was das an dieser Stelle noch nutzt. Weiterhin wird vor dem Laden eines COMMAND.PRG der Cursor nicht mehr eingeschaltet.

Neue Systemvariablen

Es sind einige neue „legale“ Systemvariablen hinzugekommen, die ohne weiteres benutzt werden dürfen, da deren Adressen auch bei zukünftigen TOS-Versionen gleich bleiben.

$51A ist ein dritter 'memvalid'-Wert. Er wird genau wie 'memvalid' ($420) und 'memval2' ($43A) beim Ermitteln der Speicherkonfiguration gesetzt (hier auf $5555AAAA). Wenn beim Reset alle drei Variablen ihren festen „Magic-Wert“ noch enthalten, wird nicht mehr der ganze Speicher gelöscht, und die Speicherkonfiguration wird übernommen. Wozu diese dritte Variable gut sein soll, ist unklar.

Die BIOS-Funktionen Bconstat, Bconin, Bcostat und Bconout werden jetzt über indirekte Sprünge aufgerufen. Die Sprungvektoren liegen für die vier Funktionen in der obigen Reihenfolge ab $51E, $53E, $55E bzw. $57E. Dabei sind bei jeder Funktion 8 (statt bisher 6) Geräte erlaubt. So springt das BIOS z. B. bei einem Bconin (2) über den Vektor $546. Die neuen Geräte 6 und 7 tun nichts, denn die Vektoren zeigen auf Leer-Funktionen. Durch die Einführung dieser Vektoren erleichtert sich z. B. die Programmierung von Drucker-Spoolern o. ä. Außerdem besteht die Möglichkeit, neue Geräte in eigenen Programmen zu definieren.

Änderungen des Desktop (AES)

Auf dem MEGA ST mit Blitter besitzt der DESKTOP einen Menüeintrag, mit dem der Blitter wahlweise ein- oder ausgeschaltet werden kann. Beim normalen Atari sieht der DESKTOP bis auf die Copyrightzeile genauso aus wie beim TOS 1.0. Die Änderungen, die wir bisher festgestellt haben, sind folgende:

Beim Kopieren eines Files auf gleicher Ebene in sich selbst (klingt kompliziert, passiert aber leicht, wenn man ein ICON ein bißchen bewegt und

wieder losläßt, so daß der DESKTOP kopieren will) war bisher das File zerstört. Bei der neuen Version bleibt das File erhalten.

Die Pfeile, mit denen die Slider bewegt werden, besitzen jetzt eine automatische Wiederholfunktion, wenn man mit der Maus auf sie zeigt und den linken Mausknopf gedrückt hält. Diese Änderung ist auch bei anderen Programmen, die mit Fenstern arbeiten, aktiv, so daß wahrscheinlich der Eventmanager des AES jetzt etwas anders funktioniert.

Die witzigen Fehler, wie Aufhängen des Event Managers, wenn man ein Icon schnell mit Doppelklick in die Menüzeile schiebt oder die Umschaltung in den vergrößerten 6★6 Font bei Benutzung eines angemeldeten Laufwerks mit der Kennung „§" (Paragraph) sind noch mit in die neue Version gewandert.

Verbessert wurde wahrscheinlich die Auswertung der Rechteckliste beim Redraw. Der Redraw erfolgt im allgemeinen flüssiger als bisher.

Die Benutzung eines Unterstrichs ist jetzt bei allen Dialogboxen ohne Probleme möglich. Je nachdem, wie ein Edit-Feld definiert war, führte es bisher zum Absturz.

Ein Wermutstropfen zum Schluß: Beim Arbeiten mit dem neuen TOS wurde ein selbst geschriebener Harddisktreiber verwendet (der von Atari fragt zwecks Erweiterung der Speicherliste das TOS-Datum ab). Innerhalb dieser Zeit wurde dreimal das Inhaltsverzeichnis einer Partition mit dem Inhaltsverzeichnes einer normalen Diskette aus dem Laufwerk A: überschrieben. Die Ursache ist bisher noch nicht geklärt, die Wirkung war die Zerstörung einer Woche Arbeit. Bleibt zu hoffen, daß dies nicht an dem neuen TOS liegt...

Alex Esser/Oliver Joppich (a little bit)

```
FC2ED8  RS232-Routinen
        fehlen teilweise

;neue Routinen
;schnelles Speicherlöschen
;Aufruf: GEMDOS-pload, VDI-clear workstation
FC4B7C MOVE.L  $0004(A7),A0           ;Anfangsadresse
FC4B80 MOVE.L  $0008(A7),A1           ;Endadresse
FC4B84 MOVEM.L D3-D7/A3,-(A7)
FC4B88 MOVEQ   #$00,D1                ;8 Register löschen
FC4B8A MOVEQ   #$00,D2
FC4B8C MOVEQ   #$00,D3
FC4B8E MOVEQ   #$00,D4
FC4B90 MOVEQ   #$00,D5
FC4B92 MOVEQ   #$00,D6
FC4B94 MOVEQ   #$00,D7
FC4B96 MOVE.W  D7,A3                  ;wird auf L erweitert
FC4B98 MOVE.L  A0,D0
FC4B9A BTST    #0,D0
FC4B9E BEQ     $FC4BA2                ;-> gerade Startadresse
FC4BA0 MOVE.B  D1,(A0)+               ;1 Byte löschen, jetzt gerade Adresse
FC4BA2 MOVE.L  A1,D0
FC4BA4 SUB.L   A0,D0                  ;Länge
FC4BA6 AND.L   #$FFFFFF00,D0          ;ganze Pages
FC4BAC BEQ     $FC4BDA                ;-> weniger als 1 Page
FC4BAE LEA     $00(A0,D0.L),A0        ;Ende des Bereichs ganzer Pages
FC4BB2 MOVE.L  A0,A2
FC4BB4 LSR.L   #8,D0
FC4BB6 MOVEM.L D1-D7/A3,-(A2)         ;256 Byte löschen
FC4BBA MOVEM.L D1-D7/A3,-(A2)
FC4BBE MOVEM.L D1-D7/A3,-(A2)
FC4BC2 MOVEM.L D1-D7/A3,-(A2)
FC4BC6 MOVEM.L D1-D7/A3,-(A2)
FC4BCA MOVEM.L D1-D7/A3,-(A2)
FC4BCE MOVEM.L D1-D7/A3,-(A2)
FC4BD2 MOVEM.L D1-D7/A3,-(A2)
FC4BD6 SUBQ.L  #1,D0
FC4BD8 BNE     $FC4BB6                ;-> nächste Page
FC4BDA CMPA.L  A0,A1                  ;Rest bis Bereichsanfang löschen
FC4BDC BEQ     $FC4BE2                ;-> fertig
FC4BDE MOVE.B  D1,(A0)+
FC4BE0 BRA     $FC4BDA
FC4BE2 MOVEM.L (A7)+,D3-D7/A3
FC4BE6 RTS
;
;Default-Systemzeit aus Uhrenchip übernehmen
;Aufruf: Reset-Routine
FC4BE8 BSR     $FC4C0C                ;Uhrencip vorhanden ?
FC4BEA BCS     $FC4C08                ;-> nein: fertig
FC4BEC BSR     $FC4C44                ;Zeit aus Uhrenchip lesen
FC4BEE CMP.L   #$FFFFFFFF,D0
FC4BF4 BEQ     $FC4C08                ;-> Fehler: fertig
FC4BF6 MOVE.W  D0,$0075B0             ;Lower Word nach System-Uhrzeit
FC4BFC SWAP    D0
FC4BFE MOVE.W  D0,$008840             ;Upper Word nach System-Datum
FC4C04 MOVEQ   #$00,D0                ;OK
FC4C06 RTS
FC4C08 MOVEQ   #$FF,D0                ;Fehler
FC4C0A RTS
;
;Test ob Uhrenchip vorhanden
FC4C0C MOVE.W  #$FC20,A0              ;Adresse Uhrenchip
FC4C10 MOVE.B  #$09,$001B(A0)
FC4C16 MOVE.W  #$0A05,D0
FC4C1A MOVEP.W D0,$0005(A0)
FC4C1E MOVEP.W $0005(A0),D1
```

Buchbesprechung



ATARI ST
Einführung in CAD
Liesert – Linden
Data Becker Buch mit Diskette
Preis: DM 69,–

CAD: Drei Buchstaben, die die Computerwelt derzeit in Atem halten. Noch vor wenigen Monaten hörte ich von dem Vertreter einer Firma der mittleren Datentechnik die Bemerkung, daß man sich über CAD-Systeme unter 50.000,– DM wohl nicht zu unterhalten brauche. Wenn man die Leistungsfähigkeit professioneller CAD-Systeme betrachtet, mag der Mann wohl recht gehabt haben, aber was sich derzeit für den Atari-ST-Computer auf diesem Gebiet tut, ist auch nicht von Pappe.

In dieser Situation weckt ein Buch mit diesem Titel natürlich großes Interesse. Man denkt vielleicht eher an eine grundlegende theoretische Einführung, aber genau das ist falsch. Auf 24 Seiten wird der Begriff CAD erläutert. Das reicht gerade für eine Erläuterung von der Bedeutung der drei Buchstaben, und worin sich ein CAD-System von einem Malprogramm a la DEGAS unterscheidet. Der Rest des Buches (265 Seiten) ist die Beschreibung eines einfachen CAD-Programms, das in GFA-Basic geschrieben ist und auf der Diskette geliefert wird, als Source Code, versteht sich. Die Beschreibung beginnt dann aber mit den notwendigen mathematischen Grundlagen der Vektorrechnung (ernsthaft interessierte Leser nehmen vielleicht doch einmal ein Schullehrbuch über Vektorrechnung in die Hand?) und geht dann schnell zur Anwendung auf CAD-Objekte und deren Manipulation über. Notfalls lassen sich auch diese 25 Seiten noch zum Bereich der Einführung zählen. Ab jetzt aber haben die GFA-Basic-„Freaks" das Wort und behalten es bis zum Schluß. Es werden alle benötigten Grundfunktionen des CAD-Programms konkret anhand des GFA-Basic-Codes erläutert. Das hat für den Anfänger natürlich den Vorteil, daß er ohne große theoretische Vorkenntnisse direkt mit der Nase auf das Problem gestubst wird und die Lösung anhand des Codes nachvollziehen kann. Getrübt wird das praxisnahe Ko[illegible]t durch die Tatsache, daß die erläuternden Texte sehr allgemein gehalten sind und selten den konkreten Lösungsweg des Listings erläutern. Das Listing erhält man somit gleich zweimal; einmal im Buch und einmal auf Diskette. Vielleicht hat der Vortrag ja an die Anfänger gedacht, die noch keinen Drucker oder GFA-Basic besitzen, jedenfalls werden im Buch etliche Seiten mit dem Listing gefüllt das auf der Diskette ebenfalls vorhanden ist. Das Programm selbst ist ein einfaches, aber funktionstüchtiges CAD-Programm, dessen Bedienung nach etwas Eingewöhnung keine Probleme bereitet. Es ist bei Fehlbedienung nicht ganz „wasserdicht", aber das ist bei einem experimentellen System nicht ganz so wichtig. Das Programm bietet halbautomatische Bemaßung, eine Auflösung von 1/10 mm, bis zu 10 Ebenen, Zeichnungen bis DIN A3, die Verwendung von Symbolen und Varianten und die Ausgabe auf einem Matrixdrucker.

Wer GFA-Basic besitzt, kann dem Listing jedenfalls viele Anregungen für eigene Programmierarbeit entnehmen. Für Liebhaber anderer Sprachen wird das Listing nicht so effektiv sein. Die grundlegenden Techniken sind aber natürlich übertragbar. Die im Anhang A gegebene Kurzanleitung ist recht hilfsreich. Für eigene Erweiterungen, zu denen das Buch ausdrücklich anregt, ist die im Anhang B abgedruckte Variablenliste sehr nützlich. Anhang C enthält ein Prozedurenverzeichnis, dessen Nützlichkeit darunter leidet, daß die Verfasser viele Prozeduren mit dem vielsagenden Namen **AKTION**, gefolgt von einer Nummer, verwendet haben. Und was bitte verbirgt sich hinter Prozedur **AKTION27**? Das darf man dann im Listing suchen, denn dort allein ist der Zweck einer Aktion kommentiert. Die alphabetische Ordnung wurde außerdem offensichtlich von einem dummen Sortierprogramm erledigt, denn auf AKTION1 folgt AKTION10 und auf AKTION19 folgt AKTION2. Wie gesagt, ist das Programm in GFA-Basic geschrieben und damit auf diesen Interpreter festgelegt.

Da aber GFA-Basic nicht zur Standardsoftware des ST gehört, wäre es zumindest sinnvoll gewesen, den Run-Only-Interpreter beizulegen.

Ein Leser ohne GFA-Basic ist jedenfalls nicht in der Lage, das Gelesene zu erproben, es sei denn, er hat aus der PD der ST-Computer eine Run-Only-Interpreter-Version (was wohl im allgemeinen der Fall sein wird). Wollte man hier etwa Lizensgebühren sparen? Jedenfalls muß in aller Deutlichkeit gesagt werden, daß dieses Buch für Leser ohne Zugang zum GFA-Basic wenig wertvoll ist da das GFA-Basic weder mit dem Standard Atari Basic noch mit dem von Microsoft-Basic kompatibel ist.

Zumindest hätte eine compilierte Version des Programms auch den Lesern, die noch keinen GFA-Basic-Interpreter ihr eigen nennen können, das Mitlesen erleichtert. Grundsätzlich ist Lernen am Beispiel zu begrüßen. Aber die Aussage heißt Lernen am Beispiel, nicht Beispiel und Lernen. Es wäre gut, die Erläuterung des Programms zu vertiefen und dafür das Listing zu [illegible]n, da man es ja doch auf Diskette hat. Und ohne GFA-Basic-Interpreter, den man ja zum Lesen des Source Codes braucht, kann man nicht viel mit dem Buch anfangen. Aber wo sonst bekommt man den Quellcode für ein funktionsfähiges CAD-Programm für 69 Mark?



LISP auf PCs
Methoden und Techniken der symbolischen Datenverarbeitung.
F. Haugg, S. Omlor.
Carl Hanser Verlag,
München und Wien, 1987.
DM 48,–

Mit zunehmender Rechnerleistung im Bereich der PCs rückt das Stichwort KI (Künstliche Intelligenz) immer mehr in den Vordergrund. Gerade als Leser der Zeitschrift ST bekommt man hierzu ja auch etwas geboten. Folglich wächst das Interesse an Literatur zu Sprachen und Anwendungen aus diesem Bereich. War der Einsteiger bisher auf englischsprachige Literatur angewiesen, so sind im Verlauf des letzten Jahres einige interessante deutschsprachige Neuerscheinungen auf den Markt gekommen. Brandneu (das Vorwort der Autoren datiert vom Januar 1987) ist das oben genannte Buch, das sich mit der Sprache LISP beschäftigt, deren Public-Domain-Version ja schon viele Anhänger gefunden hat.

Der Titel deutet es bereits an: Die Zielgruppe der Autoren ist der PC-Benutzer. Für den PC gibt es eine ganze Reihe kommerzieller LISP-Versionen. Die im Buch erläuterte Syntax bezieht sich auf MuLISP der Firma Microsoft. Erfreulicherweise orientiert sich MuLISP wie auch XLISP und LISPAS am COMMON-LISP-Standard, so daß der Leser des Buches die verwendeten Programme in weiten Teilen auf den LISP-Versionen für den ST übernehmen kann. Hardwareabhängige Programme habe ich erwartungsgemäß nicht gefunden.

Das Buch ist mit einem Umfang von 228 Seiten kein Lehrbuch, das der Lernende als alleinige Informationsquelle in Sachen LISP betrachten wird. Vielmehr ist es als Einsteigerbuch gedacht, also für Leute, die bereits Erfahrung mit herkömmlichen Sprachen wie BASIC, PASCAL oder C haben. Erstaunlicherweise beginnt das Buch aber nicht mit einem einführenden Kapitel über LISP. Völlig verblüfft nimmt der Leser die Vorteile von PROLOG in sich auf! Ein Fehldruck? Nein! Bei diesem Buch handelt es sich um den ersten Band einer Serie über Sprachen der KI, mit dem die Autoren dem Leser wohl schon vorab neugierig machen und ihre Aktualität beweisen wollten. Dann erst kommt der erwartete Überblick über die historische Entwicklung von LISP als Symbolmanipulationssprache. In diesem ersten Kapitel findet man auch eine kurze, aber einprägsame Übersicht über die verschiedenen Programmierstile. Erst im zweiten Kapitel geht es langsam los mit LISP. Die Autoren beschränken sich auf die elementaren Listenoperationen und die Erläuterung von Rekursion und Iteration. Dadurch strotzt das Buch zwar nicht von Themenmannigfaltigkeit, bringt aber dem angepeilten Interessentenkreis von LISP-Anfängern die notwendige Konzentration auf das Wesentliche in aller Ausführlichkeit. Störend wirkt in diesem Kapitel nur an einigen Stellen die Erläuterung des MUSTAR-Editors für MuLISP, mit dem der ST-Benutzer natürlich nichts anfangen kann. Das dritte Kapitel schließlich bringt einige größere Anwendungen auf dem PC in LISP programmiert. Die Beispiele Sortieren und Texttaschenrechner sind für den Anfänger ganz lehrreich. Richtig interessant sind aber die beiden nun folgenden Themen: Graphen als Listen und ein Interpreter für Structured Analysis Modelle. Umso enttäuschender ist deshalb auch die nur unvollständige Dokumentation des Source Codes für den SA-Interpreter. Das Thema Graphen als Listen ist in LISP auch deshalb interessant, weil dieses Thema sich in PROLOG besonders einfach gestaltet.

Fazit: Das Buch ist interessant geschrieben und konzentriert sich auf die für Anfänger interessanten Fragen beim Einstieg in die Sprache LISP. Das dritte Kapitel hat allerdings auch fortgeschrittenen LISP-Programmierern etwas zu bieten. Ergänzt wird der Band durch ein etwas knapp geratenes Literaturverzeichnis und eine Tabelle der MuLISP Funktionen, leider ohne Kurzbeschreibung. Das abschließende Stichwortverzeichnis ermöglicht schnelles Auffinden gesuchter Begriffe. Man darf auf den zweiten Band über PROLOG gespannt sein.

(Dr. K. Sarnow)



5300 Bonn 1
Schumannstr. 2
0228/222408

FLOPPYSTATIONEN
3.5" für ATARI ST Computer, anschlußfertig im Gehäuse mit Netzteil 1 x 720 KB 399 DM, 2 x 720 KB 799 DM
5.25" 40/80 Spur umschaltbar 360/720 KB 499 DM

RAM-AUFRÜSTUNGEN
auf 1/2/2.5/4 Mbyte, einschließlich Einbau mit 1/2 Jahr Garantie 199/998/998/1798 DM (Stand 1.8.87)

ATARI - COMPUTER
520 STM nur 579 DM / 1040 STF ab 1098 DM / MEGA ST a.A

DRUCKER		MONITORE	
NEC P6	1198 DM	ATARI SM 124/125	469 DM
EPSON LX-800	649 DM	PHILIPS Color 8833	
SEIKOSHA SL-80 AI	898 DM	12 MHz, 600 Punkte	699 DM

Schneller Reparaturservice in eigener Werkstatt !

Atari ST Erweiterung

Scanner
passend zum Programm STAD **DM 165,-**

Steckplatzerweiterung
für 2 Steckplätze umschaltbar **DM 79,-**

Roland Vodisek Elektronik
Layout-Entwicklung und Platinenfertigung
Kirchstr. 13 · 5458 Leutesdorf
Tel: 0 26 31 / 7 24 03

Wir schützen Ihre Daten

vor Mißbrauch unbefugter Dritter auf allen ATARI ST-Computern als Accessory oder PRG nach dem z. Z. weltweit als sicherst geltenden Blockschlüsselung-Algorithmus DEA 1, dem

DATA ENCRYPTION STANDARD
nach ISO und ANSI Standard

KRYPTO-STAR® ist ein Softwarepaket, welches unter Verwendung des DES-Algorithmus mit einem acht Bit cipher feedback, Daten, selbstentwickelte und gekaufte Software in eine völlig unbrauchbare und nicht mehr zu identifizierende Form umsetzt.

KRYPTO-STAR® verschlüsselte Daten sind erst mit Kenntnis eines 64 Bit-Schlüssels und einem zusätzlichen 64 Bit-Initialisierungswert zu entschlüsseln. Ohne diese Werte ist es nicht möglich, die unbrauchbaren Daten und Software in Ihren Ursprungszustand zurückzuversetzen.

KRYPTO-STAR® arbeitet nicht mit einem üblichen Passwortschutz, sondern verschlüsselt Daten Byte für Byte.

KRYPTO-STAR® ist auch DFÜ-fähig mit **KRYPTO-CONVERT®**.

KRYPTO-STAR® bietet somit das höchste Maß an Datensicherheit für Jedermann, welches bisher nur einer kleinen Gruppe vorbehalten war.

KRYPTO-STAR® ist in Betrieb selbsterklärend und somit kinderleicht zu nutzen.

KRYPTO-SOFT erstellt auch individuelle Sicherheits-Systeme auf Anfrage.

KRYPTO-STAR®
DM 98,-* (incl. Handbuch)
KRYPTO-CONVERT®
DM 35,-* (incl. Beiblatt)

BESTELL-CHECK
Hiermit bestelle ich
☐ **KRYPTO-STAR®** zum Preis von **DM 98,-***
☐ **KRYPTO-STAR®** Handbuch vorab **DM 25,-*** (wird bei Kauf von **KRYPTO-STAR®** angerechnet).
☐ **KRYPTO-CONVERT®** zum Preis von **DM 35,-***
System:.................
Lieferung per
☐ Scheck
☐ Nachn. (+ DM 10,- Gebühr)

KRYPTO-SOFT GmbH
Weizenfeld 36, D-5060 Berg. Gladbach 2
Tel. 02202/30602

Der Unterschied ist deutlich...

NEU: Ab sofort alle Produkte ohne Kopierschutz!

Update Aktion:
G RAMDISK II oder
INTERPRINT ohne Ramdisk — INTERPRINT mit Ramdisk **DM 50,—** (für Harddiskbesitzer)
AS SOUND SAMPLER — AS SOUND SAMPLER II **DM 70,—**

Alle anderen Produkte erhalten Sie in der neuesten Version und ohne Kopierschutz gegen DM 20,—.
Updates werden gegen Einsendung der Originaldiskette und des Betrages bar/Scheck bearbeitet.

G RAMDISK II DM 49,—

- Resetresident mit 4 MB
- Größe frei wählbar
- Laufwerke C – E
- abschaltbar
- integrierter Druckerspooler

INTERPRINT DM 99,—

- integrierte Ramdisk, resetfest wie G RAMDISK II, jedoch C – P
- Universeller Druckertreiber
- Anpassung aller Drucker
- Hardcopy für alle Matrixdrucker (9, 18 und 24 Nadeln)
- integrierter Druckerspooler auch für Hardcopies

INTERPRINT ohne Ramdisk DM 49,—

AS Sound Sampler Software	DM 79,—
AS Sound Sampler II Software	DM 149,—
AS Sound Sampler Hardware	DM 129,—
AS Sound Sampler II Hardware	DM 249,—
2 Demodisketten	DM 10,—

DISK HELP DM 79,—

- Reparatur defekter Disketten
- Lesefehler werden korrigiert.
- Nicht mehr lesbare Files werden rekonstruiert.
- Directories und FAT werden wieder lesbar.
- Nur bei Schreib-/Lesefehlern, nicht bei physikalischen Diskettenschäden

G DISKMON II DM 98,—

- Uneingeschränkte Bytemanipulation auf der gesamten Diskette
- Darstellung in ASCII, Dezimal und Hexadezimal
- Kopieren, suchen, ersetzen, Tracks formatieren u.v.m.
- komfortabler Bootsektoreditor
- Schnelldruck von Sektoren
- Vollständig interaktiv
- Schnelle Diskettenreparatur bei read error

Harddisk Help & Extension DM 128,—

- Backup-Programm, sichert Ihre Harddiskdaten auf Disketten
- Partition Backup
- Extrem schnell
- Selected Filecopy auch für 720 KB Files
- Tree zeigt alle Directoryeinträge.
- Selected Tree selektiert nach beliebigen Kriterien.

Achtung: G RAMDISK II, INTERPRINT, DISK HELP, G DISKMON II, Harddisk Help & extension und AS Sound Sampler sind von der Gesellschaft der unabhängigen EDV Berater e.V. empfohlen.

G DATA 

Siemensstr. 16 · 4630 Bochum 1
TELEFONISCHE BESTELLUNGEN: 023 25/608 97

Schweiz: Senn Computer AG
Langstr. 21 · CH-8021 Zürich

Bestell-Coupon
Bitte senden Sie mir: ☐ kostenloses Informationsmaterial
zzgl. DM 5,— Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße
Ort



KMAX-

– 7,5 Millionen Instruktionen pro Sekunde auf dem ST

Eine Transputerkarte für den Atari: Darauf haben die Rechenleistungsfanatiker gewartet. Ein 32-Bit-Transputerchip vom Typ INMOS T414 sorgt für absolute Höchstleistungen. Wie im Konzept der Transputer vorgesehen, ist auch der Anschluß zusätzlicher Chips für theoretisch unbegrenzte Geschwindigkeit völlig problemlos. Aber was ist eigentlich ein Transputer? Keine Angst, bevor wir näher auf die KMAX-Erweiterung von Kuma Computers aus England eingehen, soll zunächst über Transputer und die dazugehörigen Konzepte, zum Beispiel RISC, berichtet werden.

I. Was ist ein Transputer?

Ein Transputer ist im Grunde ein Mikroprozessor. Seine Architektur orientiert sich im wesentlichen an zwei Konzepten. Zunächst wäre das RISC-Konzept zu nennen. RISC bedeutet 'Reduced Instruction Set Computer', also 'Computer mit reduziertem Befehlssatz'. Um die Bedeutung dieses Konzeptes zu verstehen, muß man sich einiges aus der Prozessorarchitektur und der Geschichte der Mikroprozessoren vergegenwärtigen.

Zuerst zur Historie. Die ersten Mikroprozessoren waren recht einfach konzipiert. Sie enthielten nur einige wenige Befehle, die insgesamt auch nicht besonders mächtig waren. Die Entwicklung ging in der Folgezeit einen ähnlichen Weg wie bei der Programmiersprache Basic. Man fügte immer mehr Befehle hinzu, die zum Teil recht komplexe Aufgaben mit einem Befehl erledigen. Ein anderes Beispiel ist die Sprache ADA, die im amerikanischen Verteidigungsministerium entwickelt wurde. Die Komplexität hat aber einige Nachteile: Zuerst einmal wird ein Interpreter, der eine große Anzahl von verschiedenen Befehlen ausführen muß, zwangsläufig langsamer als eine kompaktere Version. Auch der Befehlsdecoder eines Mikroprozessors ist im Prinzip eine Art Interpreter, der durch einen großen Befehlssatz langsam wird. Zum anderen zeigt sich in der Praxis, daß die hochkomplexen, dadurch natürlich auch sehr speziellen Befehle nur selten genutzt werden. Im Gegenteil, es kann bei entsprechenden Strukturierungsmöglichkeiten (zum Beispiel bei einer Hochsprache) durch Modularisierung und hierarchisches Konzept fast immer günstiger und einfacher sein, selbst ein entsprechendes Modul aus einfachen Befehlen zusammenzubauen. Ein weiteres Problem der Befehlsvielfalt, an dem besonders ADA krankt, ist die Schwierigkeit, sich all diese Befehle auch zu merken, was nun einmal notwendig ist.

Aber zurück zur Historie. Die Mikroprozessoren wurden immer komplexer und schneller. Dennoch zeigten sich Grenzen der konventionellen Technik. Anlaß für diese Entwicklung war der Glaube, daß die Entwicklung von Compilern für Hochsprachen durch möglichst große Befehlsvielfalt erleichtert würde. In den letzten Jahren stellte sich das jedoch als Irrtum heraus. Die Compilerbauer verwenden, wie statistische Untersuchungen zeigen, viel lieber die einfachen Grundbefehle. Überlegungen zur Prozessorarchitektur zeigen darüber hinaus, daß es erheblich effizienter ist, Interpreter, also Decoder, für einfache Maschinensprachen zu bauen. Komplexe Befehlssätze erfordern nicht nur deshalb mehr Zeit, weil bei der Interpretierung mehr Befehle berücksichtigt werden müssen, sondern auch deshalb, weil für komplizierte Befehle sogenannte Mikroprogrammsteuerungen verwendet werden müssen. Ein Mikroprogramm ist ein prozessorinterner Ablauf von Steuerbefehlen, deren einzelne Bits direkt die ausführenden Teile des Prozessors ansteuern. Diese Befehle sind in einem internen ROM gespeichert. Auch der Befehlssatz des 68000 ist auf diese Weise implementiert. Der Vorteil dieses Verfahrens ist die verhältnismäßige Einfachheit, mit der der Befehlssatz einer CPU bei der Herstellung geändert werden kann. Der große Nachteil ist, daß Mikroprogrammsteuerungen auf dem Chip sehr viel Platz in Anspruch nehmen. Dieser Platz steht nicht mehr für andere Dinge wie Register, schnelle Rechenwerke, Cache-Speicher und ähnliches zur Verfügung. Außerdem besteht ein Mikroprogramm selbst für einen einfachen Befehl oft aus mehreren Schritten, wobei jeder Schritt einen Taktzyklus benötigt. Dadurch wird die Ausführungszeit erhöht.

Bei einem speziellen, einfachen Befehlssatz kann man beide Probleme umgehen. An die Stelle der Mikroprogrammsteuerung tritt eine Hardware, die die Decodierung der Befehle direkt in einem Schritt erledigt. Bis zu einer gewissen Anzahl von Befehlen ist dieses Verfahren schneller und platzsparen-

der als Mikroprogramme. Da die Größe einer Dekodierungshardware aber leider nicht nur linear mit dem Befehlssatz anwächst, ist man tatsächlich sehr eingeschränkt. Den durch Hardware-Steuerungen gewonnenen Platz kann man zum Beispiel für eine große Anzahl von Registern verwenden. Das hat den Vorteil, das die Zahl der zeitraubenden Speicherzugriffe reduziert werden kann, weil mehr Programmteile in Registern gehalten werden können. Die einfachen Befehle können auch in kürzester Zeit abgearbeitet werden. Typische RISC-Architekturen haben daher weit unter 100, oft nur zwei Dutzend Befehle und manchmal über 1000 Register. Die Anzahl der Instruktionen pro Sekunde ist dadurch bei einem solchen Prozessor extrem hoch. Compiler arbeiten auf RISC-Maschinen oft extrem effizient und produzieren einen Code, der kaum noch zu optimieren ist.

Haben Sie bitte Verständnis dafür, wenn diese Einführung in das RISC-Konzept stark vereinfacht ist. Leider haben wir nicht genügend Platz, uns diesem Thema mit der gebührenden Präzision zu widmen.

Die RISC-Idee ist in der Transputer-Konzeption auf eine recht eigenwillige Art verwirklicht. Zwar hat der Prozessor eine relativ geringe Anzahl von Befehlen (111 – für eine 'klassische' RISC-Architektur sind es sogar noch recht viele), die Abarbeitung erfolgt jedoch nicht über eine Hardware-Steuerung. Auch besitzt der Prozessor nur sechs Register, was der RISC-Philosophie entgegensteht. Vor allem gibt es außer dem User-Stackpointer nur drei Allzweckregister, die intern als eine Art Mini-Stack fungieren. Dafür sind allerdings auf dem Chip 2 KByte extrem schnelles statisches RAM implementiert, das die Register recht gut ersetzen kann. Es gibt nur eine einzige Adressierungsart; mehr als zwei Adressierungsarten besitzt praktisch keine RISC-Maschine. Zum Vergleich seien hier die Daten einer typischen RISC-Architektur genannt. Es handelt sich dabei um die RISC-II der Berkeley-Universität: 39 Befehle, 2 Adressierungsarten, fast alle Befehle benötigen nur einen Taktzyklus, 140 Register, Hardware-Steuerung. An dieser Stelle soll zum Thema RISC nicht mehr gesagt werden, obwohl es noch vieles zu diesem Konzept der Zukunft anzumerken gäbe.

Das zweite Konzept, das man bei der Entwicklung der Transputer berücksichtigte, stammt aus der theoretischen Informatik. Es handelt sich hierbei um das Prozeß-Modell, in dem jeder Ablauf als Prozeß dargestellt wird, der durch Nachrichten mit anderen Prozessen kommunizieren kann. Ein Prozeß wird durch eine ankommende Nachricht angestoßen und gibt nach Beendigung eine Nachricht zurück. Selbstverständlich kann ein Prozeß auch weitere Prozesse anstoßen. Auch eine Summe von Prozessen kann wieder als Prozeß aufgefasst werden. In den Transputern ist das Prozeß-Modell höchst interessant verwirklicht: Jeder Chip verfügt über vier superschnelle bidirektionale serielle Schnittstellen, die mit bis zu 20 Megabaud senden und empfangen können. Diese Schnittstellen können für Nachrichten im Sinne des Prozeßmodells verwendet werden. Arbeitet nur ein Transputer, wirkt das ganze Konzept nicht besonders aufregend. Spannend wird es aber, wenn man an die Verschaltung einer ganzen Anzahl solcher Chips denkt. Wie gesagt, man kann auch eine ganze Reihe von Prozessen zusammen wieder als einen Prozeß auffassen. Ein Programm, das auf einem Transputer läuft, kann man sich also als eine Reihe von Prozessen vorstellen, die untereinander transputerintern kommunizieren. Das gleiche Programm könnte aber auch auf mehreren Transputern laufen; jedem Transputer, der über seinen eigenen privaten Speicher verfügt, wird ein Prozeß zugeordnet. Die seriellen Schnittstellen, die übrigens als 'Links' bezeichnet sind, übernehmen dann die Rolle des Mediums für die Übertragung von Nachrichten zwischen den einzelnen Prozessen. Besonders effizient kann dieses Konzept mit der von INMOS speziell für diesen Zweck entwickelten Sprache OCCAM ausgenutzt werden (andere Hochsprachen wie C, Pascal oder Fortran sind ebenfalls möglich). OCCAM erlaubt die Programmierung von Parallel-Verarbeitung (in einer der nächsten Ausgaben wird OCCAM ausführlicher besprochen werden).

Damit ist das Zauberwort gefallen. Parallele Ausführung verschiedener Programmteile zur Geschwindigkeitssteigerung ist das zentrale Konzept aller Rechnerkonzepte neuerer Generationen. Viele Probleme sind bereits in ihrem Konzept hervorragend geeignet für parallele Bearbeitung, so zum Beispiel die meisten Rechenaufgaben im Bereich Computergrafik. Mit Transputern kann man leicht ein Rechnersystem konzipieren, das die Aufgabe optimal erfüllt. Dennoch kann das komplette Programm genausogut auf einem Transputer entwickelt werden, der gesamte Prozeß läuft dann zwar langsamer, aber es funktioniert. Die Übertragung auf ein anderes System erledigt der Compiler. Natürlich muß der Systemprogrammierer sich selbst Gedanken machen, welche Rechnerstruktur seinem Problem am besten entspricht. Viele mathematische Probleme lassen sich sehr effizient mit einer 'Pipeline' aus hintereinandergeschalteten CPUs lösen, andere wiederum mit einer würfelförmigen Schaltung. Alle diese Schaltungen können mit Transputern mit extrem niedrigem Aufwand erstellt werden. Man braucht keine komplizierten Bussysteme wie bei herkömmlichen Multiprozessorsystemen. Die Transputer können mit minimal zwei Drähten miteinander verbunden werden. Es existieren Spezial-Transputer für die Steuerung von Disk-Drives, die vollständig in ein Transputer-Netzwerk integriert werden können. Den Vogel schießt allerdings ein Floating-Point-Transputer ab, der Fließkommamathematik im 64-Bit-IEEE-Format mit bis zu 1.5 Megaflop (das sind Fließkommaoperationen pro Sekunde!!!) erledigt. Der Floating-Point-Transputer ist übrigens pin- und softwarekompatibel mit der 'normalen' Ausführung. Leider sind die Chips noch sehr teuer. Falls der Preis einmal fällt, ist der Austausch problemlos.

Mit Transputern sind bereits Rechner mit mehreren 100 solcher Bausteine geplant, deren Geschwindigkeit in der absolut höchsten Leistungsklasse liegt.

II. Die KMAX-Transputerkarte

Atari-Benutzer können sich trotz der hohen Preise für die Transputerchips und die dazugehörige Software des sogenannten Transputer Development Systems mit Hilfe der Kuma KMAX-Transputerkarte schon jetzt in diese Zukunftstechnik einarbeiten. Für 995 Pfund, das sind etwa 3000 DM, erhält man ein kleines gelbes Kästchen, das eine Transputerkarte zum Anschluß an den ST enthält. Diese Karte ist, wie bereits erwähnt, mit einem Transputer des Typs T414 sowie 256KByte RAM bestückt. Der T414 ist ein echter

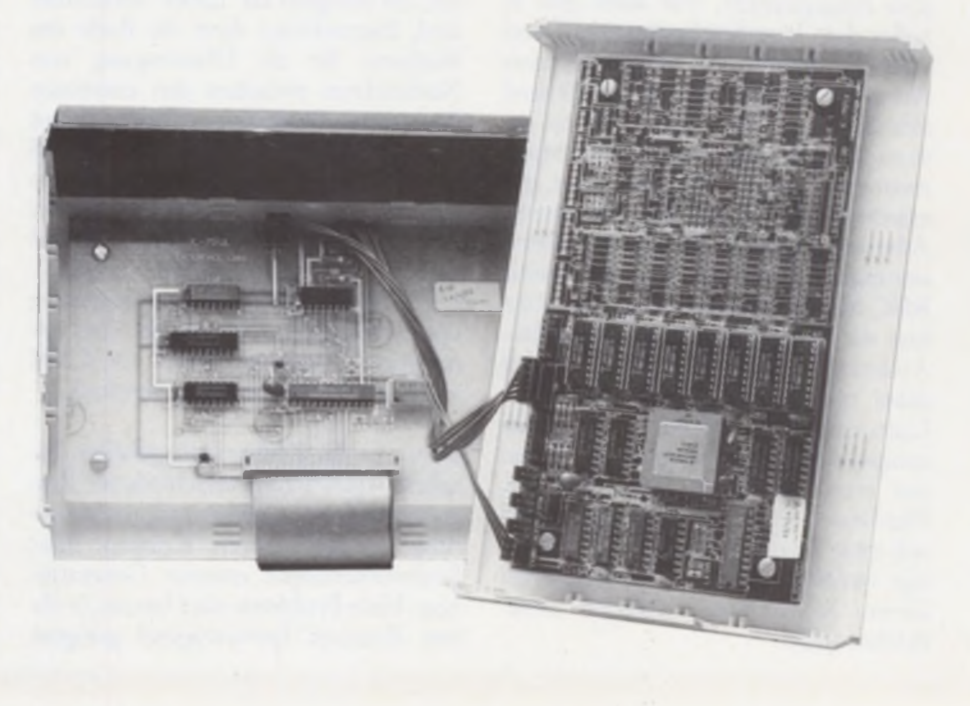
32-Bit-Prozessor. Er wird mit 15 MHz getaktet, was einer Rechenleistung von 7,5 Millionen Instruktionen pro Sekunde entspricht. 2 KByte RAM sind im Prozessor enthalten. Auf der Karte ist Platz für einen zweiten Transputer mit 256 KByte RAM. Leider sind die Fassungen für diese Erweiterungen noch nicht auf die Platine gelötet. Die Garantie für das System bleibt nur erhalten, wenn die Erweiterung von Kuma durchgeführt wird.

Der Anschluß der KMAX-Karte erfolgt am Romport über ein Flachbandkabel. Im KMAX-Gehäuse ist eine Interfacekarte enthalten, die Daten, die über einen Parallelport vom Atari geliefert werden, mit Hilfe eines sogenannten Linkadaptors in das INMOS Link-Format übersetzt. Auf diese Weise ist es möglich, mit Hilfe spezieller Software über einen der vier Links mit dem Transputer in Verbindung zu treten und ihn mit Software zu versorgen. Dabei erweist sich das von INMOS vorgesehene Boot-Verfahren als nützlich: Nach einem RESET wartet jeder Transputer solange, bis über einen der Links eine bestimmte Datenfolge gesendet wird, die der Transputer als Programm erkennt, empfängt und schließlich ausführt. Alternativ ist von INMOS auch das Booten von einem ROM vorgesehen, was aber auf der KMAX-Karte nicht möglich ist. Die Transputer-Erweiterung muß also immer vom Atari mit Software gefüttert werden, was für ein Entwicklungssystem auch unbedingt sinnvoll ist. Da der Transputer immer unter Kontrolle des Atari steht, können abgestürzte Transputer-Programme mit Hilfe einer RESET-Leitung immer gestoppt und analysiert werden, obwohl der Atari für laufende Transputer-Programme auch Terminal spielt. Man muß für einen RESET des Transputers nur das Terminalprogramm unterbrechen und die RESET-Leitung für den Transputer setzen. Ganz einfach und sicher, das unangenehme Neubooten des Atari bei Softwarefehlern entfällt. Nachdem jetzt die technische Seite des Transputeranschlusses erklärt ist, soll die mitgelieferte Software erläutert werden. In dieser Hinsicht ist der Lieferumfang des Kuma-Transputersystems eher bescheiden. Im Gegensatz zum Original-INMOS-Transputer-Development-System, zu dessen Lieferumfang ein OCCAM II-Compiler und eine ganze Reihe von Hilfsprogrammen gehört, ist im Kuma-System im Augenblick nur ein Cross-Assembler enthalten. Dies ist allerdings ein recht leistungsfähiges und komfortables Programm.

Wahrscheinlich wird bei Erscheinen dieses Tests zusätzlich auch ein OCCAM-Compiler, der von Metacomco geschrieben wurde, zur Verfügung stehen. Daß dieser bereits den vollen OCCAM II-Leistungsumfang besitzen wird, ist jedoch unwahrscheinlich.

Zu dem RISC-Processor-Development-System, wie Kuma die KMAX-Erweiterung im Untertitel leicht übertrieben nennt, wird außer dem Assembler eine 52seitige, englische Dokumentation im DIN A4-Format geliefert. In diesem Handbuch wird der Assembler ausführlich beschrieben, außerdem werden alle wichtigen Informationen über die Implementierung des Transputer-Anschlusses an den ST gegeben. Es sollte damit kein Problem sein, auch von eigenen Programmen aus Kontakt mit dem Transputer zu bekommen. Weitere Kapitel des Manuals beschäftigen sich mit dem Datenformat der INMOS-Links, sowie allem weiteren Wissenswerten über die konkrete Hardware-Realisierung der eigentlichen Transputerkarte.

Einige Modifikationen können auf der Karte noch vorgenommen werden; z. B. kann der 15-MHz-Transputer problemlos gegen die schnellere (und teurere) 20-MHz Version ausgetauscht oder die Geschwindigkeit der Links verändert werden. Über das Transputer-Konzept sollte man allerdings zumindest in groben Zügen Bescheid wissen. Die Dokumentation geht darauf nämlich überhaupt nicht ein. Es ist also notwendig, sich entsprechende Literatur von INMOS zu besorgen. Selbstverständlich kann man an den Transputer auch wie an ein normales Mikroprozessor-System herangehen und die besonderen Spezialitäten, die in seinem Konzept enthalten sind, zuerst einmal übergehen. Damit geht dann allerdings der eigentliche Sinn eines solchen Entwicklungssystems, nämlich die Einarbeitung in eben diese Konzepte, völlig verloren. Kuma hätte ruhig ein wenig mehr über die Transputer-Konzepte ins Handbuch schreiben können. Es fragt sich allerdings, ob man in Assembler der besonderen Stärke der Transputer, nämlich der extremen Vorbereitung für Parallelverarbeitung, auch nur anäherungsweise gerecht werden kann. Eigentlich erfordert dieses Transputerkonzept doch eine etwas mächtigere Sprache, die selbst ebenfalls auf dem Konzept paralleler Prozesse beruht; eben das bereits erwähnte OCCAM. In Verbindung mit anderen, nicht-parallelen Sprachen wie C, Pascal oder Fortran, für die OCCAM, wenn man nicht vollständig in dieser Sprache programmieren will, den kontrollierenden Rahmen abgibt, ist es wohl erst möglich, sinnvolle Anwendungen auf einem Transputer zu schreiben und diese dann auf Transputer-Netzwerke zu übertragen. Auch das Prozeß-Konzept von Modula II könnte auf Transputer-Systemen äußerst interessant sein.



ATARI ST

steckbar steckbar
Speichererweiterungen
auf **1 MByte** 260/520 STM **275,–**
auf **2,5 MB / 4 MB** a.A.
260 ST, 520 STM, 1040 STF

Jede Erweiterung einzeln im Rechner getestet!
Sehr einfacher Einbau. **Ohne Löten einbaubar!!!**
Gut bebilderte, ausführliche Einbauanleitung.
Vergoldete Mikro – Steckkontakte für MMU-Adapter. Optimale **Schonung des MMU-Sockels**
Achten Sie auf **Mikro-Steckkontakte**
Kein Bildschirmflimmern!!! Ohne zus. Software, ohne zus. Stromversorgung. Test in ST 4/86.

ECHTZEITUHR 129,–

Jede Uhr im Rechner getestet! Einbau durch Einstecken in den Rechner. Freier ROM Port – für spätere Erweiterungen. **Uhrzeit, Datum, Wochentag Anzeige.** Dank Lithium-Batterie ca. **10 Jahre Laufzeit. Hohe Genauigkeit.**
Schaltjahrerkennung.
Bei Verwendung von GFA Basic-Modul keine Bootdiskette erforderlich!
Komplettsysteme preisgünstig!

Händleranfragen erwünscht.

WEIDE-ELEKTRONIK

Regerstr. 34 · 4010 Hilden · Tel.: 0 21 03/4 12 26
Kostenloses Info!!!

Schweiz	Niederlande	Belgien
SENN Computer AG	Commedia	MICRO-CONNECTION
Langstr. 31	1" Looiersdwarsstr. 12	St Katelijnevest 18
CH-8021 Zürich	1016 VM Amsterdam	2000 Antwerpen
Tel. 01/241 73 73	Tel. 020/23 17 40	Tel.: 03/231 15 40



NLQ NLQ NLQ

- Aufrüstsatz für alle
Epson **MX, RX, FX, JX** Drucker
Apple Macintosh Drucker
Emulation für (FX & JX)
Viele Features! Info anfordern!
FX **199,–** MX **179,–** RX 80 **149,–**

GFA-BASIC V2.0 MODUL 198,–

3,5" Laufwerke

Disketteneinzelstation (1 MByte)	**398,–**
Diskettendoppelstation (2 MByte)	**698,–**
im PC Gehäuse 2 Laufwerke á 1 MB	**749,–**

CO-Prozessor 68881 890,–

inkl. Software für Megamax C, Mark Williams C, DRI C, Prospero Fortran 77, Modula II, CCD Pascal ⊕
erhöht die Rechengeschwindigkeit z.T. um Faktor 750: einfachster Einbau, rein steckbar - ohne Löten!
unbedingt **ausführliches** INFO anfordern!
Trackball statt Maus (platzsparend u. schnell) **119,–**

The Price War Is Over. We Won.

Z. B. ANWENDERPROGRAMME:

Art Director	129,– DM
BS-Fibu	999,– DM
BS-Handel	799,– DM
Film Director	129,– DM
GfA-Basic Interpreter	144,– DM
GfA-Basic Compiler	144,– DM
Pro Sound Designer	169,– DM
Signum!	359,– DM
STAD	159,– DM
T.I.M. Buchhaltung	179,– DM

ODER SPIELE:

Airball	72,– DM
Alternate Reality	62,– DM
Arkanoid	38,– DM
Barbarian	66,– DM
Flight Simulator II (M+F)	129,– DM
Gauntlet	59,– DM
Goldrunner	72,– DM
Guild of Thieves	69,– DM
Roadrunner	69,– DM
Sub Battle Simulator	69,– DM

Gerald Köhler
Soft- und Hardware für Atari ST

Mühlgasse 6
6991 Igersheim

Tel. 0 79 31 / 4 46 61
(24h-Service)

Natürlich führen wir noch weitaus mehr Produkte für den Atari ST. Fordern Sie deshalb unseren Gratiskatalog an, es lohnt!

ODER HARDWARE:

10 Disketten 3,5 Zoll 1DD	27,90 DM
10 Disketten 3,5 Zoll 2DD	32,90 DM
Diskbox 3,5 Zoll (80 ST.)	22,90 DM
Quickshot II	13,50 DM
Quickshot II+	22,90 DM
Staubschutzhaube 520 ST	22,90 DM
Druckerständer DS-80	27,90 DM
AS-Soundsampler (mit SW)	199,– DM
PAL-Interface (alle ST)	258,– DM
Speichererw. auf 2,5 MB	848,– DM

ODER PD-SOFTWARE:

- Riesenauswahl aus 250(!) Disks mit mehr als 1000 Programmen.
- außerdem alle "ST-Computer"-Disks sofort lieferbar.
- einseitige + doppelseitige Kopien.
- Einzelprogramm-Service.
- Paket- und Staffelpreise.
- "Sound Sampler"-Service.
- Gratiskatalog (mehr als 20 Seiten) anfordern, Sie werden staunen!

Neueröffnung · Neueröffnung · Neueröffnung · Neueröffnung · Neueröffnung · Neueröffnung

LaSch *das Buch und Software Haus · Inh. Rainer Langner u. Franz Schnitzler GbR*
Nohlstraße 76 · 4200 Oberhausen 1 · ☎ 02 08 / 80 90 14

Die Ersten 100 Personen die einen Softwarekatalog bei uns anfordern (Rückporto DM 1,30) bekommen eine Diskette G R A T I S dazu!
ATARI **Public-Domain-Software** ab DM 5,– auf FUJI-FILM Disketten

Neueröffnung · Neueröffnung · Neueröffnung · Neueröffnung · Neueröffnung · Neueröffnung

---

DIE SYMBOLBIBLIOTHEKEN FÜR GFA DRAFT PLUS

PNEUMATIK - HYDRAULIK
ca. 340 SYMBOLE
149,– DM

ELEKTRONIK
ca. 450 SYMBOLE
149,– DM

MÖBEL
ca. 400 SYMBOLE
149,– DM

ZEICHENSÄTZE u. DEMOZEICHNUNGEN
je 49,– DM

WEITERE INFORMATIONEN: Ralf Hille Datentechnik · Mittelstr. 61 · 4322 Sprockhövel 2 · Tel. 02339/7191 o. 6028
MESSETERMINE: 4.9 - 6.9 Koblenz · 30.10 - 1.11 Ludwigshafen · 13.11 - 15.11 Recklinghausen

Das Problem der Assembler-Programmierung auf Transputern wird zusätzlich dadurch erschwert, daß INMOS nicht unbedingt begeistert auf die Absicht der Assemblerprogrammierung ihres Lieblingskindes reagiert. Es ist sehr schwierig, vom Hersteller überhaupt Informationen über den Befehlssatz der Transputerchips zu erhalten, was zur Folge hat, daß das Handbuch den Befehlssatz auch nur sehr vage beschreibt. Man hat also eigentlich nur zwei Möglichkeiten: Entweder man probiert alles aus, wie es die Entwickler von Kuma in der Anfangsphase des Projektes auch tun mußten, oder man versucht, sich von INMOS die Originalunterlagen zu beschaffen, was – wie man hört – ein schwieriges und langwieriges Unterfangen sein soll. Der Befehlssatz der Transputer unterteilt sich in zwei Gruppen: Es gibt 16 einfache Befehle, deren Funktion in der Dokumentation recht gut und mit ausreichender Ausführlichkeit beschrieben ist. Dazu gibt es noch knapp hundert weitere Befehle, die nur ohne weitere Erklärungen dem Namen nach mit ihren Operatoren aufgelistet sind. Nicht einmal eine Erklärung über das, was sie tun, ist dabei. Natürlich sind viele der Befehle selbsterklärend, so zum Beispiel 'add' für addieren. Bei anderen Befehlen ist es dagegen notwendig, etwas mehr über die Struktur des Transputers zu wissen.

Am besten schaut man sich die mitgelieferten Beispiele, einige einfache und zum Teil beeindruckend schnelle Programme, sorgfältig an, um mehr über den Transputer-Assembler zu erfahren.

Alles in allem ist das Handbuch recht brauchbar. Trotzdem wird man um zusätzliche Informationsquellen nicht herumkommen. Außerdem wird sich wohl jeder, der halbwegs ernsthaft mit dem Transputer arbeiten will, so schnell wie möglich einen OCCAM- oder sonstigen Compiler zulegen. Die Assembler-Programmierung ist wohl wirklich nur für den Einstieg oder, später, für extrem zeitkritische Prozeduren gedacht. Meines Erachtens läßt sich einfach alles, was an Transputern über normale Mikroprozessoren hinausgeht, in einer Hochsprache logischer und effizienter, auf gewisse Weise 'natürlicher' programmieren.

III. Der Cross-Assembler

Bei diesem Programm handelt es sich um ein integriertes Maschinensprache-Entwicklungspaket, das außer dem Assembler einen recht komfortablen Editor, einen Debugger, einen Zeilen-Disassembler und alle Routinen, die zur Programmübergabe an den Transputer benötigt werden, enthält.

Es handelt sich um eine TOS-Applikation. Nach dem Laden meldet sich der Assembler ähnlich wie der Kommandointerpreter mit dem Prompt 'XPA>' und wartet auf Kommandos. Im Handbuch wird an Hand von Bedienungsbeispielen sehr übersichtlich in das Programm eingeführt. So hat man rasch ein erstes kurzes Programm erstellt, das sich mit irgendeiner Meldung auf dem Atari-Bildschirm bemerkbar machen soll. Doch wie kommt der Transputer an den Atari-Bildschirm?

Um Daten an den Atari zu übergeben, hat Kuma eine Bibliothek einfacher Terminalroutinen mitgeliefert. Diese Bibliothek liegt im Sourcecode vor, kann also analysiert und vielleicht sogar verstanden oder modifiziert werden. Um diesen Sourcecode in eigenen Programmen zu verwenden, setzt man im Editor einfach den Cursor an die Stelle, an der der einzufügende Text erscheinen soll, und führt ein Kommando zum Einlesen von Text aus. Dabei wird der Text automatisch in den alten Text eingefügt. Durch Diskettenoperationen wird der alte Sourcecode immer nur erweitert, nie einfach überschrieben. Um ein neues Programm einzuladen, muß erst der alte Text gelöscht werden. Ist ein Programm fertiggestellt, wird durch die Taste 'a' der Assembler aufgerufen. Dieser ist mit 50.000 Instruktionen pro Minute recht flott und schreibt den produzierten Sourcecode gleich automatisch in den Speicher des Transputers. Dabei ist die Speicheradresse frei wählbar. Standardmäßig wird im internen Speicher des Transputer-Chips begonnen, der erheblich schneller als der restliche Speicher auf der Platine ist. Dabei wird ausreichend Platz für den User-Stack reserviert. Sobald man mit der Taste 'g' ein Programm im Transputer startet, schaltet sich der Assembler in einen Terminal-Modus, in dem der Transputer mit Hilfe der Bibliotheksroutinen auf die Resourcen des Atari zugreifen kann. Falls das Transputer-Programm nicht ganz das Gewünschte tut, kann es jederzeit durch Drücken von Control-A gestoppt werden. Dabei wird der Transputer in einen speziellen Analyse-Modus gesetzt, der dem Assembler die vollständige Kontrolle über den Chip erlaubt. Es ist auch möglich, lediglich den Terminalmodus zu verlassen, indem man Control-C betätigt.

Die gesamte Arbeitsweise ist praktisch, effizient und schnell. Der Editor erlaubt Blockoperationen und führt diese auch schnell aus, der Assembler arbeitet formatfrei, daß heißt, man kann den Text nach Geschmack formatieren, auch mehrere Anweisungen (durch Kommas getrennt) in einer Zeile sind möglich. Die üblichen Pseudo-Anweisungen für die Definition von Datenbereichen sind vorhanden, Labels und Konstanten können definiert werden. Eine spezielle Direktive erlaubt es, bei Verwendung mehrerer Transputer einen bestimmten Code-Abschnitt an einen bestimmten Transputer zu senden. Nach dem 'g'-Kommando führt dann jeder Transputer den ihm zugeordneten Code aus. Dabei ist es völlig gleichgültig, wie, wo und über wieviele Ecken, sprich über wieviele und welche Links, der Transputer angeschlossen ist. Man kann also dem Assembler die Struktur eines Netzwerkes aus Transputern angeben, die Übersendung der Programme an die entsprechenden Chips erledigt er automatisch.

Der Debugger erlaubt die Inspektion und Veränderung beliebiger Speicherbereiche des Transputers. Mit einem besonderen Kommando kann der Disassembler aufgerufen werden, der bei jedem Aufruf jeweils 16 Zeilen Code auf den Bildschirm bringt. Wegen einer Besonderheit der Transputer-Assemblersprache ist es dabei möglich, das der Disassembler-Code anders aussieht als das, was in den Assembler eingegeben wurde. Der Assembler erlaubt nämlich die Abkürzung einiger mühsamer Formulierungen dieser Assembler-Sprache, die der Disassembler dann unabgekürzt zurückübersetzt.

Besonders wichtig für potentielle Anwender, die sich in die Transputertechnik einarbeiten wollen, ist natürlich die Geschwindigkeit einer solchen Erweiterung. Maximal 7,5 Millionen Instruktionen kann der T414-Transputer in seiner 15-MHz-Ausfüh-

rung bewältigen. Verwendet man die 20-MHz-Version, sind es sogar bis zu 10 Millionen Instruktionen, mit dem Floating-Point-Transputer sogar bis zu 1,5 MFlop bei 64-Bit-IEEE-Fließkommazahlen. Genügen 32 Bit Genauigkeit, steigt die Geschwindigkeit dieses Chips auf mehr als das doppelte.

Leider haben die auf der KMAX-Karte verwendeten Transputerchips noch kleinere Fehler, so daß der 256-KByte Speicher auf der Transputerplatine nur mit einer großen Anzahl von Wait-Zyklen angesprochen werden kann. Die volle Geschwindigkeit erreicht die Karte also nur mit Programmen, die das interne 2K-RAM des Chips ökonomisch nutzen. Ein mitgeliefertes Primzahlenprogramm, das auf dem Sieb des Eratosthenes basiert, erreicht auf der Karte eine Rate von mehr als 1.000.000 Primzahlen pro Minute (52 Sekunden für 1 Million), bei Ausgabe jeder 50.000sten Zahl. 2.000.000 32-Bit-Integer-Zahlen werden in 8 Sekunden miteinander multipliziert. Dabei werden die Zahlen im Speicher gehalten und nicht in Registern. Kuma selbst hat für KMAX in Benchmarks eine ungefähr 5-fache Geschwindikkeitssteigerung gegenüber dem ST ermittelt. Das dürfte ungefähr der Geschwindigkeit eines 68020 ohne Arithmetik-Coprozessor entsprechen.

Die Geschwindigkeit dieses oder überhaupt eines einzelnen Transputers ist jedoch im Grunde gar nicht so interessant.

Ihre besonderen Stärken entfalten Transputer eben nur in Gruppen – nach dem Motto: Eine starke Gemeinschaft... Man kann eben für nahezu jede Anwendung eine geeignete Hardware-Struktur mit vergleichsweise geringem Aufwand realisieren, deren Leistungsfähigkeit kaum durch das Konzept begrenzt ist. Wichtig an der KMAX-Erweiterung ist, daß es für umgerechnet 3000 Mark möglich ist, sich relativ preiswert in eine echte Zukunftstechnologie, nämlich die der parallelverarbeitenden Computer einzuarbeiten. Die Preise für das Original-Inmos-Entwicklungspaket sind eben doch sehr hoch, auch wenn man die bisher bessere Softwareunterstützung in Betracht zieht. Bei Kuma ist man aber sehr mit der Softwareentwicklung beschäftigt – ein OCCAM-Compiler wird, wie bereits erwähnt, zum Erscheinungsdatum dieses Testes wahrscheinlich zur Verfügung stehen. Kurz vor Redaktionsschluß erreichten uns auch etwas nähere Angaben zu dem OCCAM-System, das übrigens von Metacomco entwickelt wird: Es wird zwei Versionen geben; eine OCCAM-ST-Version, die OCCAM-Programme für den 68000er des ST compiliert und in England inclusive Mehrwertssteuer knapp 60 Pfund kosten wird, sowie ein Cross-Compiler-System, das Ihre Programme auf dem ST für die Transputerkarte compiliert. Der ST-Compiler wird in diesem Paket ebenso enthalten sein wie ein Programm, das die Verteilung von OCCAM-Programmen auf mehrere Transputer erlaubt. Dieses 'große' Paket wird unter 200 Pfund kosten. Alle Compiler werden vorerst nur den OCCAM-I Leistungsumfang unterstützen, die OCCAM-II Version ist noch nicht soweit. Sobald entsprechende Software zur Verfügung steht, ist das KMAX-Paket ein echtes Transputer-Entwicklungssystem, auf dem auch die Entwicklung großer Transputer-Anwendungen möglich sein dürfte.

(CS)

Public-Domain Service

Updates

U1, U2...

Manchmal kommt es vor, daß Autoren ihre Programme verbessern und uns Updates schicken. Natürlich aktualisieren wir die Disketten sofort, aber umtauschen bzw. neu bespielen ist leider nicht möglich. Damit Sie eine aktualisierte Diskette (die Sie vielleicht schon besitzen) nicht noch einmal bestellen müssen, haben wir uns folgendes überlegt:

Wir werden, falls erforderlich, jeden Monat spezielle Update-Disketten zusammenstellen, damit Sie auf einen Schlag alle verbesserten Programme erhalten können und nicht alle geänderten Disketten gesondert bestellen müssen. Natürlich werden gleichzeitig auch die Hauptdisketten aktualisiert. Somit erhalten Sie von uns immer die neusten Versionen.

Wir hoffen, daß dieses System für Sie die einfachste Lösung bietet.

Der Kopierservice nach dem alten Verfahren wird wegen mangelnder Nachfrage und zu großen Aufwandes ab dieser Ausgabe eingestellt.

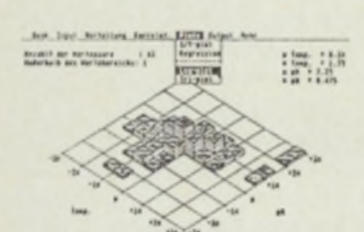
Ihre ST-Computer-Redaktion



Die Neuheiten dieses Monats

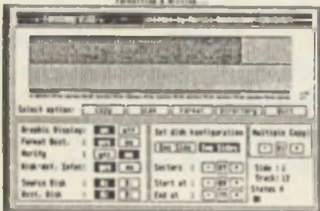
Diskette 76
Schulungssoftware

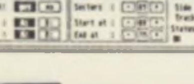
● *Periode:* Das Periodensystem der Elemente. Chemielernprogramm mit vielen Optionen. Enthält die Daten jedes chemischen Elements (s/w). ● *Statistik:* Grafische Darstellung und Datenauswertung (s/w).



Diskette 77
Utilitydisk

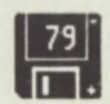
● *FCOPY 2.0:* Schnelles Kopierprogramm. Formatieren, Verify, Multiple Copy, Scan und Teilkopie. ● *Speeder:* Verdoppelt Schreib- und Lesegeschwindigkeit. ● *Filecopy:* Kopieren von Einzeldateien. Auch mit einer Floppy. ● *Formater:* Erhöht die Diskkapazität. ● *Fastformat:* Formatiert Schnelladedisks. Durch besondere Anordnung der Sektoren wird immer schneller geladen. ● *Taskcopy: Diskbackup im Multitasking.* ● *Profiramdisk: Resetfeste Ramdisk mit Ausführungsprofil.*





Diskette 78
Utilitydisk

● *AESLIB:* Bibliothek für GFA-Basic. Enthält komplette AES-Bibliothek. Mit Linker, der jede benötigte Prozedure automatisch zum Programm hinzulädt. ● *LINE_A:* Line_A Binding für C-Programme. ● *Conv.WP:* Konvertiert ASCII-Dateien in den WP-Code von 1st Word. Dadurch Umbruch, Formatieren etc. möglich. ● *Player:* Programm zum Abspielen von Interruptmusik incl. Source in C. ● *Musik:* Ähnliches Programm aber als Accessoir. ● *Setz_Uhr:* Programm zum Stellen der Zeit + Datum incl. Source in Assembler.



Diskette 79
Etikettendruck

● *Label Expert:* Druckt alle Art von Etiketten. 3,5 Zoll- und 5,25 Zoll-Disketten, Audio-Kasetten, Video-Kasetten, Ordner u.a. (s/w). ● *Grafikett:* Bedruckt Disklabels auch mit Grafik, die aus beliebigem Grafikbild ausgeschnitten wird (s/w). ● *Etikettdat:* Druckt Adressenaufkleber (s/w). ● *Etikette.GFA:* Luxuriöses Disklabeldrucker. Mit vordefinierter, artbezogener Grafikeinbindung (s/w).



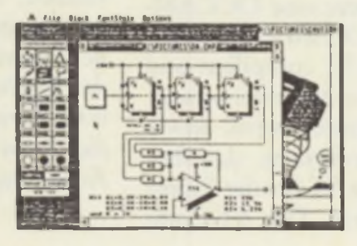
Diskette 80
Spieledisk

● *U_Boot:* Ziel dieses Spiels ist es, U-Boote im Meer mit eigenen Sonden zu orten und ihnen auszuweichen (s/w). ● *Ärger:* Eine Umsetzung des wohl bekanntesten Brettspiels 'Mensch ärger dich nicht' auf den ST (s/w).



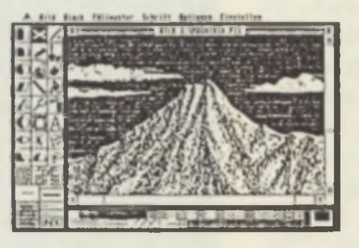
Diskette 81
Malprogramm

● *Public Painter:* Malprogramm mit interessanten Fähigkeiten. Verarbeitet z. B. gepacktes Format oder Bilder von SNAPSHOT. Eingebaute Lassofunktion zum Ausschneiden beliebiger Teilbereiche. Blockmanipulation ähnlich EASYDRAW. Vergrößern, Verkleinern, Drehen, Biegen, Verzerren, Überschneiden. Drucken von Bildausschnitten (s/w).



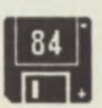
Diskette 82
Malprogramm

● *Masterpainter:* Läuft in allen drei Auflösungen. Konvertiert jede Auflösung in beliebige andere. Blockbearbeitung. Vergrößern, Verkleinern, Rotieren etc., bis zu 24 Screens. Ausschnittdruck. ● *Little Painter:* Malprogramm mit P6-Treiber. Rasteroperation, Musterspray, Verwischen, Drehen, Biegen, Blockladen und -speichern, Fontlader (s/w).

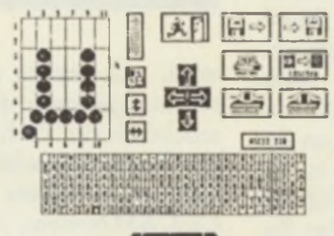


Diskette 83

● *Starofix:* Komfortabler Zeichsatzeditor für Down-Load Zeichen. Für Star NL10 und kompatible. Ein Zeichensatz wird per Autostartprogramm in den Drucker geladen. Ausdruck mit jedem Programm mit neuen Zeichen. Viele Fonts schon vorhanden. ● *Starlist:* Ausdruck von ASCII-Dateien mit Sonderzeichenwandlung.



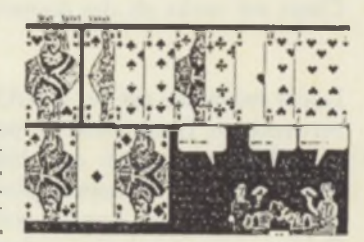
Diskette 84
Druckedit: Zeicheneditoren für 1st Word-Druckeranpassung. ● *Draft Pin:* Die Zeichen werden per Maus editiert und dann automatisch in beliebiges .HEX-File eingebaut. Ideale Ergänzung zu 1st Word bzw. Wordplus (s/w).





Diskette 85
Spieledisk

● *Skat:* Eine Runde SKAT gefällig? Sie spielen gegen zwei vom ST gesteuerte Zocker. Ansprechende Grafik (s/w). ● *Delta:* Gehirnmuskeltraining. Ähnlich dem Zauberwürfel (f). ● *Grusel:* Grafik- und Sounddemo. In der Geisterstunde ist auf dem Friedhof allerhand los. Skelette, Äbte und Gespenster treiben ihr Unwesen (f).



Updates

Die Updates dieses Monats wurden auf zwei Disketten zusammengefasst.

U1
● *Haushalt (44) Video (73) Lager (44) Van_gogh (68)*

U2
Edimax (55): Eigener Zeichensatz, eigene Fileselectorbox, jump to error, schnelleres Scrolling (nur s/w). ● *Tosedit (55):* Kürzer, schneller, Blockbefehle. ● *Termin (46):* Einige Verbesserungen.

Die Update-Disks können Sie zu den gleichen Bedingungen wie die normalen PD-Disketten erhalten.

Sonderdisks

Die folgenden Programme sind nicht Public-Domain.
Sie können aber bei uns bezogen werden.

A.) – *TOS:* Die letzte Disketten-Version des TOS vom 6.2.1986. Zum fehlerfreien Betrieb des Rechners benötigen Sie diese Version, da z. B. mit älteren Versionen manche Ordner nicht gelesen werden (0 Bytes free).

Unkostenbeitrag 15,– DM incl. Diskette.

B.) – *RCS:* Das Resource Construction Set aus dem ATARI-Entwicklungspaket. Unverzichtbar bei der Erzeugung von Objektbäumen, Dialogboxen, Drop-Down Menüs. Kurzum: Ein Muß für GEM-Programmierung.

Unkostenbeitrag 15,– DM incl. Diskette.

Bis auf die Höhe des Unkostenbeitrags gelten die gleichen Versandbedingungen wie bei der Public-Domain-Software.



Public-Domain Service

1. Diskette
● *Neochrome V.06*: Malprogramm [f] ● *Doodle*: Malprogramm [s/w]

2. Diskette
● *Joshua*: Diskmonitor – CP/M Emulator (CP/M 80) ● *Megaroids*: Weltraumspiel ● *Diskformatter*: erweitert Diskkapazität auf 399 K bzw. 809 K ● *Kartei-Kasten*: luxuriöses Karteiprogramm ● *50/60 Hz*-Umschaltung für Monitore

Diskette 3 und 4 mit Volksforth 83 wurden durch die Disketten 49, 50, 51 ersetzt.

5. Diskette
● *Ramdisk* mit Source-Code ● *SG 10* Druckertreiber als Desk-Accessory ● *JAM Term*: Terminalprogramm ● *Diskcopy* ● *ST Grafic*: bewegte Grafiken in 3D ● *CP/M Emulator V 8.4*: neueste Version

6. Diskette
Diese Diskette beinhaltet einige Nutzprogramme für den CP/M-Emulator. Lauffähig mit Version 8.4.
● *CPMTOS*: TOSCPM wandelt CP/M-ASCII-Files in TOS-Format und zurück ● *SCOPY*: Kopierprogramm für ein Laufwerk ● *DDTZ*: Debugger ● *SETKEY*: Tastaturumbelegung, u. v. m.!!

7. Diskette
● *XLISP 1.71* der Interpreter zum ST-Kurs Künstliche Intelligenz

8. Diskette
● *Diskmon*: ● *GfA Basic-Demo*: ● *Kuvert*: zum Bedrucken von Kuverts ● *Mini-CAD*: ein praktisches Programm mit Source-Code ● *Vokabel*: Vokabeltrainer ● *Superformat*: unterschiedliche Diskformate

9. Diskette
● *Funkplot*: Funktionsplotprogramm incl. Funktionseingabe über Tastatur, Integration und Differenzation ● *Konvert*: Konvertierung von Farb- auf Monochrombilder in 3,6 Sekunden ● *Spriteed*: ein kleiner Spriteeditor ● *Super*: ein Superhirnspiel [f]

10. Diskette
● *Iconeditor* ● *Puzzlepuzzle*: Puzzleprogramm ● *Krabat-Schach* Schach für den ST

11. Diskette
● *PROLOG 10* Die Sprache der neuen Generation. Begleitsoftware zum Kurs 'künstliche Intelligenz' incl. 95 KByte Anleitung.

12. Diskette
● *Rechner*: ein luxuriöser Taschenrechner
● *Calc*: Taschenrechner als Desk-Accessory
● *Biorhythmus*: Erläuterung überflüssig

13. Diskette
● *Pascalshell*: GEM-Menüsteuerung für ST-Pascal ● *MMCOPY ACC*: Filecopy als Accessoir ● *Happy 4*: 'Vier gewinnt' ● *Labyrint ACC*: Labyrint als Accessoir ● *MAD ACC*: Ihr Rechner entwickelt Eigenleben ● u. a.

14. Diskette
● *Tempelmon*: Monitor von Th. Tempelmann [s/w] ● *Transistorvergleicher*: Transistordatenverwaltung ● *Bitte ein Bit*: Bitcopyprogramm ● *Diskmon* ● *ViFibu*: Kleine Finanzbuchhaltung ● *EX 06*: Fängt Systemfehler und zeigt sie an.

15. Diskette
Druckeranpassung 1st_Word
Auf dieser Diskette befinden sich alle uns zur Verfügung stehenden Druckertreiber. Es sind folgende Treiber enthalten:

ASCII	Epson RS80	Epson_U
Epson_LX80	IBM Komp	Gabi 9009
IBM	MP_165	KX-P1092G
LQ_800	NL10_IBM	NEC_P2&P3
NEC_P6&P7	OKI_M192	NL10_Pan
OKI_M182	TA-SE320	OLI M2
SMM804	Centronics D300	TAXAN 8105T
C 6020	Gabi 9009	Brother M 1109
Citizen MSP10	Olivetti ET 1111	Olympia Carrera

16. Diskette
Sammlungen diverser Programme incl. Source Code [s/w] ● *Etikettendruck*: bedruckt Etiketten (Basic) ● *Funkplot*: mit Funktionseingabe (Basic) ● *Kassetten- und Videodateien* unter GEM (Basic) ● *Typewriter*: Schreibmaschinenkurs (Basic) ● *LIFE*: Fortpflanzungs-Rechnung (Pascal) ● *Schiffe*: Schiffe versenken (Demo für GEM-Routinen) (Basic)

17. Diskette
Programmsammlungsdiskette incl. Source in Basic [s/w] ● *Aktien*: Der Aktienmarkt ● *Habu*: Verwaltung von Konten ● *Lohnbuchhaltung & Versanddruck* ● *Steuer*: Einkommens- und Lohnsteuerberechnung.

18. Diskette
● *Doodle plus*: erweiterte Doodle Version; incl. Bilder ● *Power*: Sonderzeichensatz [s/w] ● *Konverter*: Wandelt alle Bildformate ● *Datum*: Datum auf Disk ● *Smooth*: Scrollt eine 64 000 KB Doodle-Bild ● *Format*: erzeugt diverse Diskformate ● *Kbug*: ST leidet unter Käferbefall

19. Diskette
● *Kermit*: Stark verbesserte GEM-Version des Terminalprogramms ● *Disksort*: Druckt Diskinhalt als Liste oder als Etikette ● *Terminal*: Terminalprogramm ● *Fontedit*: Zeichensatzeditor für GEM-Draw und Easy-Draw Fonts. [s/w] ● *OKI-Init ACC*: Steuercodes für Microline 182 incl. Source-Code

20. Diskette
● *Harddisk*: Treiber für Harddisk. Automatischer Start von der Platte (keine Bootdiskette nötig) ● *Toram*: Kopiert Programme in Ramdisk ● *RSCAnalyse*: Analyse von RCS-Files [s/w] ● *DESK Uhr ACC*: Zeigt ständig Zeit und Datum an. Drei einstellbare Weckzeiten. ● *GEMTOOL*: Demo für GEM-Programmierung [s/w]

21. Diskette
● *Adress*: Private Adressverwaltung ● *Etikett*: Etikettendruck ● *Haushalt*: Privatkontenverwaltung ● *Selektor*: Dateiverwaltung (30 frei erstellbare Felder)

22. Diskette
● *Plot*: Grafische Darstellung von Wertepaaren ● *Myterm*: Terminalprogramm (GfA-Basic) ● *VKT*: Vokabeltrainer

23. Diskette
● *DGDB*: 'Das große deutsche Ballerspiel', Aktionspiel siehe RELAX 12/86 [f] ● *Kalah*: Das bekannte Brettspiel ● *Fractals*: Schnelle 'Apfelmännchen-Version' in Assembler.

24. Diskette
● *Taurus*: Strategiespiel für mehrere Spieler. Siehe Relax 12/86. Schöne Grafik, viele Optionen [f], (leider nur Mega-Rechner)

25. Diskette
● *Disk Engineer*: Diskutility (Rescratch, ...) ● *Omikron Basic*: Run-Only Interpreter des neuen Basics

26. Diskette
● *Codetabelle*: ASCII-Tabelle (Accessoir) [s/w] ● *Life_Game*: Das bekannten Generationenspiel (Accessoir) [s/w] ● *Term*: Und noch ein Terminalprogramm ● *Biorhythmus*: ST-Basic schlägt zurück [s/w] ● *Konto*: Kontoführung in GfA-Basic [s/w] ● *KX P1091*: Accessoir zur Druckersteuerung

27. Diskette
● *NEC-Treiber*: Hardcopy-Treiber für 24-Nadeldrucker (NEC und kompatible). Aufruf über ALT+HELP ● *ST-Sound Demo*: Die Siegerdemo des ST-Computer-Sound-Wettbewerbs. 'Zottl, Knurp und Schnatter hauen rein' (siehe Titelbild ST-Computer Nr. 12)

28. Diskette
● *GfA Basic Programme incl. Source* ● *Funkplot*: [s/w] ● *Terminal*: Terminalprogramm in GfA [s/w] ● *Konverter*: Konvertiert ST-Basic in GfA-Basic ● *ST-Kill*: Utilityprogramme ● *Sprite-Editor* [s/w]

29. Diskette
● *Diskutility*: Der Weg ans Directory (manipuliert Löschschutz, erzeugt waschechtes IBM-Format, und mehr) [incl. Source in Pascal] ● *Adress_pas*: Adressverwaltung, die einige käufliche Programme das Fürchten lehrt. ● *ST-Graph*: 3-D Balkengrafik [s/w] [incl. Source in GfA]

30. Diskette: Spielesammlung
● *Bohnen* ● *Schiffe*: Schiffe versenken ● *Mastermind*: Superhirn für Superhirne [f] ● *Dragon_hunt*: Ein äußerst gefräßiger Drachen will ausgerechnet Ihnen an den Kragen [f] ● *Memo*: Memory (2 Versionen) ● *Solitaire*: Das bekannte 'Taschenspiel' ● *4 Gewinnt*: Spiel in GfA-Basic incl. Source [s/w]

31. Diskette: Spielesammlung
● *Patience*: Wunderschöne Version dieses Kartenspiels (siehe RELAX Nr. 1/87 [s/w] ● *3D-Irrgarten*: Finden Sie einen Weg aus einem dreidimensionalen Labyrinth. Trotz ST-Basic sehr schnell [s/w] ● *Avalun*: Sie verwalten ein Königreich (ST-Basic)

32. Diskette
● *Mathlib*: Eine umfangreiche Mathematikbibliothek für Pascal. Trigonometrische, Matrizenalgebra und vieles mehr, direkt zum Einbinden ● *Finanzmathe*: Berechnung von Zinsen, Laufzeiten, Laufzeit, Tilgungsplänen, etc

33. Diskette: diverse Utilities
● *Encrypt*: Ver- und Entschlüsseln von Dateien ● *Ramdisk*: Ramdisk beliebiger Größe ● *Recover*: Regeneriert gelöschte Files (!!) ● *Sterm*: VT52 Terminal + Source in Assembler ● *Freeram*: Zeigt freien Speicher

34. Diskette: Der Assembler zum Kurs
● *Assemblerpaket nach M 68000-Standard* incl. 2-Pass Assembler, Disassembler, Debugger, Editor.

35. Diskette
● *Andromeda*: Zeichenprogramm (DEGAS-ähnlich) ZOOM, Beschriften in vier Richtungen und allen Größen, Kopieren, Linienstil / Art / Breite und alle normalen Grafikbefehle [s/w] ● *3D-Konstruktion*: Zum Entwurf dreidimensionaler Gegenstände [s/w] ● *Globus*: Animierte Erdballgrafik

36. Diskette
● *CLI*: vollständiger Kommandointerpreter (CP/M, MS-DOS Oberfläche) ohne Maus. Ausführen von Batch-Files, Autostart, Kopieren auf Ramdisk, Belegung der Funktionstasten, Ausführen von Programmen (TOS + GEM), etc. ● *IBM Drive*: Ermöglicht den Filetransfer IBM <–> ST ● *Drupatch*: Zum Ausdruck von Source Files. Mit Zeichenwandlung wie 1ST Word, universell anpaßbar ● *DB_Etikett*: Druckt Etiketten aus DB-Master

37. Diskette
● *SCIENTIFIC*: Zeichensatzeditor. Erzeugt Anpassungsfile für 1st Word [s/w] ● *Convert*: Konvertiert ST- und Microsoft Basic Programme in GfA-Basic ● *Check*: Rechtschreibkorrekturprogramm. Erstellt eigenes Lexikon, Übernahme der neuen Wörter, Korrektur direkt im Text.

38. Diskette
Alle Bilder des ST-Kalenderwettbewerbs, die in Heft 1/87 veröffentlicht wurden, incl. Picture-Compactor.

39. Diskette
● *Jackpot*: Einarmiger Bandit, incl. Sourcecode in GFA-Basic [s/w] ● *Lucky*: Spielautomat [s/w] ● *Hanoi*: Türme von Hanoi

40. Diskette
● *Deluxe Fontmaster ST*: Druckprogramm zum Erstellen von Karten und Tabellen, mit vielen Zeichensätzen, Proportionalschrift und Blocksatz, eigener Text- und Zeichensatzeditor. Bis zu 10 Schriftarten in einem Text mischen und vieles mehr [s/w]

41. Diskette
● *Painter*: Zeichenprogramm unter GEM. Verfügt über einige Besonderheiten. Z. B. Zoom, Linien und Füllmustereditor, Blockoperationen (Verschieben, Drehen,...) [s/w] ● *VIKA*: Bedrucken von Kassetten und Videohüllen [s/w] ● *Kalender*: Diverse Kalenderberechnungen. U. A. Ausgabe eines Jahreskalenders auf Drucker oder Bildschirm, mit Anzeige aller Feiertage (sehr wichtig)

42. Diskette
● *FRAM 77*: Statische Berechnungen für Konstrukteure. Sehr umfangreich ● *Meßwert*: Grafische Darstellung von Meßwerten [s/w], incl. Sourcecode in Pascal

43. Diskette
● *System*: Accessoir ähnlich dem Kontrollfeld. Besonderheiten: Uhr und 4 Weckzeiten, Druckereinstellung (Rand, Schriftart,...) [s/w], incl. Sourcecode in Pascal ● *Snaphelp*: Hiermit kann SNAPSHOT (1st Word Utility) auch auf DEGAS angewendet werden, incl. Sourcecode in Pascal

44. Diskette
Spieledisk
● *TTT*: Dreidimensionales Tic Tac Toe (4 * 4 * 4) [s/w] ● *El Bozo*: El Bozos Window Puzzle ● *Senso*: Fordern Sie Ihr Gedächtnis [s/w] ● *Kalah* ● *Tri Hex*: Räumliches Tic Tac Toe (3 * 3 * 3) [s/w] ● *Börse*: Spekulieren am Börsenmarkt [s/w] ● *Solitär*: [s/w]

45. Diskette
● *Complex*: Bibliothek f. komplexe Berechnung ● *Master*: Generationsberechnung mit frei definierbaren Regeln [s/w] ● *Pegasus*: Malprogramm [s/w] ● *Con_ST*: Terminalprogramm u. a.

46. Diskette
● *Termin*: Terminplanung als Accessoir [s/w] ● *KFZ-Stat*: Statistikprogramm für's Auto [s/w] ● *Lager*: Lagerverwaltung [s/w] ● *ST_Phone*: Telefonkostenberechnung [s/w]

47. Diskette
Utilitydisk
● *M_Copy_87*: Backupprogramm mit vielen Optionen und Musik ● *Copy_FMT*: Backupprogramm als Accessoir ● *Uhrzeit*: diverse Utilities zur eingebauten Uhr ● *1st_step*: Kalender und Biorhythmus [s/w] ● *Lotto*: [s/w] ● *P2_init* u. a.

48. Diskette
● *Word 1.4*: Programm zum Erstellen eines luxuriösen Lexikons für 1st Wordplus

49. 50. 51. Diskette
● *Volksforth 83*: Die neueste Version des Forthinterpreters. Die Software zum Forthkurs. Alle drei Disketten gehören zum Forth-System.

52. Diskette
● *Molekül*: Darstellung chemischer Moleküle. In s/w und Farbe, wobei in Farbe mit 3D-Effekt. Atom, Ionen oder Van der Waals-radius. Ein Molekül kann gedreht und sogar animiert werden.

53. Diskette
● *Matrix*: Komfortables Matrizenberechnungsprogramm mit voller GEM-Steuerung (Icons, Windows, Menüzeile). Maximal 16 * 16 Matrizen, doppelte Genauigkeit (64 Bit). ● *LIB*: Sammlung von ST-Pascal-Bibliotheken. Incl. Source und Anleitung. Stringumwandlung (str <–> int, long_int, real) Datums + Uhrzeit-Rechnungen Textformatierungen (links, rechtsbündig, zentriert, u.v.m.)

54. Diskette
Spielesammlung:
● *Bounce*: Bouncing Bubbles. Schnelles Actionspiel für Joystick und Tastatur (s/w) ● *Domino*: Gefräßige Würmer auf Nahrungssuche (s/w) ● *Trio*: Verbessertes Tic-Tac-Toe mit Verschieben (s/w) ● *Disc*: Die Serie der Spielautomaten reißt nicht ab (s/w)

55. Diskette
Editorendisk
● *TOSEDIT*: Editor unter TOS. Gleichzeitiges Bearbeiten von zwei unabhängigen Texten. Reichhaltige Befehle ● *EDIMAX*: Editor unter GEM. Mit Funktionstastenbelegung. Weiterhin zeichnen sich EDIMAX durch sehr schnelles Scrolling aus.

56. Diskette
● *Ausland*: Fremdsprechentrainer Englisch mit Sprachschatz für Fortgeschrittene ● *Fugger*: Verwalten Sie ein Fürstentum und versorgen Sie ihr Volk. (s/w) ● *Yahtzee*: Altbekanntes Zockerspiel auf den ST umgesetzt.

57. Diskette
Wissenschaftliche Anwendungen
● Kepler: Grafische Darstellung von Himmelskörper-Bewegungen nach den Keplerschen Gesetzen. Variation von Masse, Radius, Zeit, potentieller und kinetischer Energie. ● *Kerne*: Der Krieg der Kerne, eine futuristische Schlacht, im Innern eines Computers suchen Rechenprogramme sich gegenseitig zu zerstören. ● *Meßwert*: Verarbeitung von Meßergebnissen und grafische Darstellung (s/w).

58. Diskette
● *SM-Konverter*: Konvertiert SM-Manager-Texte in 1st Word (Wordplus)-Texte und natürlich auch zurück. ● *Compress*: Komprimiert beliebige Files. ● *Label*: Ändern des Diskettennamens. ● *Eti-Master*: Etikettendruck zu DB-Master ● *Urlaub*: Hilft den Urlaub zeitoptimiert zu berechnen

59. Diskette
● *Themadat*: Assoziative Datenbank mit bis zu 200 Datensätzen

60. Diskette
● *Synthi*: Erzeugt div. Sounds, die abgespeichert und über 'DOSOUND' in eigene Programme übernommen werden können (incl. der Source-Code) (s/w) ● *Waschi*: Waschhilfsprogramm für Junggesellen (s/w) ● *Expert*: Ein einfaches, lernbegieriges Expertensystem ● *Adressen*: Adressverwaltungsprogramm mit Etikettendruck und einfacher Textverarbeitung

61. Diskette
● *Hacomini*: Erzeugt Hardcopies in Miniaturformat (48 mm * 68 mm) (s/w) ● *Helper*: Hilfsprogramm bestehend aus Diskmonitor und Druckerutility ● *Bildpo*: Scrollt eine Bilderkette über den Schirm. Ideal für eigene Diashows oder Werbegrafiken ● *Datobert*: Grafische Darstellung Linien- und Kuchengrafik, sowie in dreidimensionaler Balken- und Flächengrafik. Überlagerung einzelner Datenserien (s/w) ● *Direct all*: Directorydruck ● *Überweisung*: Überweisungsdruckprogramm aus ST 6/87.

62. Diskette
Midi-Diskette
● *MIDIDRUM*: Rhythmusmaschine mit leicht und vielseitig editierbaren Sounds ● *MIDIPLAY*: Erlaubt das Einspielen, Speichern und Ausgeben von Midi-Sequenzen, incl. Source-Code in GFA-Basic

63. Diskette
Spielesammlung
● *Fuffy*: Sehr schön gestaltetes Kniffel-Spiel mit zusätzlicher Auswertegrafik (s/w) ● *Snake*: Eine Schlange schlängelt sich durchs Leben, auf der Suche nach Fressen. Man achte auf Hindernisse (s/w) ● *Bomber*: Scramble-Spiel (f) ● *Reversi*: Der Name sagt alles (f).

64. Diskette
● *E-PLAN*: Konstruktionsprogramm für elektronische Schaltungen. Alle elektronischen Bauelemente sind vorhanden. Schaltung können auch beschriftet und die Bauteile in die Stückliste übernommen werden.

65. Diskette
Mal- und Zeichenprogramme
● *Columbia*: Malprogramm mit vielen Extras. Mehrere Bildschirme, Blockoperationen, Spiegeln, Verzerren, Drehen, Lupe (s/w) ● *Paintlux*: Malprogramm mit zusätzlicher Bibliotheksverwaltung. Dadurch können z. B. elektronische Bauteile aus der Bibliothek in das Bild eingefügt werden (s/w)

66. Diskette
Spieledisk
incl. Source-Code in OMIKRON Basic
● *MAZIACS*: Abenteuerspiel. Unser goldsuchender Held im Labyrinth, im Kampf mit bösartigen Spinnen und anderen Hindernissen (s/w) ● *Quadrat*: Bilderpuzzle. Ordnung im Chaos (s/w).

55. Diskette
Editorendisk
● *TOSEDIT*: Editor unter TOS. Gleichzeitiges Bearbeiten von zwei unabhängigen Texten. Reichhaltige Befehle ● *EDIMAX*: Editor unter GEM. Mit Funktionstastenbelegung. Weiterhin zeichnen sich EDIMAX durch sehr schnelles Scrolling aus.

56. Diskette
● *Ausland*: Fremdsprechentrainer Englisch mit Sprachschatz für Fortgeschrittene ● *Fugger*: Verwalten Sie ein Fürstentum und versorgen Sie ihr Volk. (s/w) ● *Yahtzee*: Altbekanntes Zockerspiel auf den ST umgesetzt.

57. Diskette
Wissenschaftliche Anwendungen
● *Kepler*: Grafische Darstellung von Himmelskörper-Bewegungen nach den Keplerschen Gesetzen. Variation von Masse, Radius, Zeit, potentieller und kinetischer Energie. ● *Kerne*: Der Krieg der Kerne, eine futuristische Schlacht, im Innern eines Computers suchen Rechenprogramme sich gegenseitig zu zerstören. ● *Meßwert*: Verarbeitung von Meßergebnissen und grafische Darstellung (s/w)

58. Diskette
● *SM-Konverter*: Konvertiert SM-Manager-Texte in 1st Word (Wordplus)-Texte und natürlich auch zurück. ● *Compress*: Komprimiert beliebige Files. ● *Label*: Ändern des Diskettennamens. ● *Eti-Master*: Etikettendruck zu DB-Master ● *Urlaub*: Hilft den Urlaub zeitoptimiert zu berechnen

59. Diskette
● *Themadat*: Assoziative Datenbank mit bis zu 200 Datensätzen

60. Diskette
● *Synthi*: Erzeugt div. Sounds, die abgespeichert und über 'DOSOUND' in eigene Programme übernommen werden können (incl. der Source-Code) (s/w) ● *Waschi*: Waschhilfsprogramm für Junggesellen (s/w) ● *Expert*: Ein einfaches, lernbegieriges Expertensystem ● *Adressen*: Adressverwaltungsprogramm mit Etikettendruck und einfacher Textverarbeitung

61. Diskette
● *Hacomini*: Erzeugt Hardcopies in Miniaturformat (48 mm * 68 mm) (s/w) ● *Helper*: Hilfsprogramm bestehend aus Diskmonitor und Druckerutility ● *Bildpo*: Scrollt eine Bilderkette über den Schirm. Ideal für eigene Diashows oder Werbegrafiken ● *Datobert*: Grafische Darstellung Linien- und Kuchengrafik, sowie in dreidimensionaler Balken- und Flächengrafik. Überlagerung einzelner Datenserien (s/w) ● *Direct all*: Directorydruck ● *Überweisung*: Überweisungsdruckprogramm aus ST 6/87.

62. Diskette
Midi-Diskette
● *MIDIDRUM*: Rhythmusmaschine mit leicht und vielseitig editierbaren Sounds ● *MIDIPLAY*: Erlaubt das Einspielen, Speichern und Ausgeben von Midi-Sequenzen, incl. Source-Code in GFA-Basic

63. Diskette
Spielesammlung
● *Fuffy*: Sehr schön gestaltetes Kniffel-Spiel mit zusätzlicher Auswertegrafik (s/w) ● *Snake*: Eine Schlange schlängelt sich durchs Leben, auf der Suche nach Fressen. Man achte auf Hindernisse (s/w) ● *Bomber*: Scramble-Spiel (f) ● *Reversi*: Der Name sagt alles (f).

64. Diskette
● *E-PLAN*: Konstruktionsprogramm für elektronische Schaltungen. Alle elektronischen Bauelemente sind vorhanden. Schaltung können auch beschriftet und die Bauteile in die Stückliste übernommen werden.

65. Diskette
Mal- und Zeichenprogramme
● *Columbia*: Malprogramm mit vielen Extras. Mehrere Bildschirme, Blockoperationen, Spiegeln, Verzerren, Drehen, Lupe (s/w) ● *Paintlux*: Malprogramm mit zusätzlicher Bibliotheksverwaltung. Dadurch können z. B. elektronische Bauteile aus der Bibliothek in das Bild eingefügt werden (s/w)

66. Diskette
Spieledisk
incl. Source-Code in OMIKRON Basic
● *MAZIACS*: Abenteuerspiel. Unser goldsuchender Held im Labyrinth, im Kampf mit bösartigen Spinnen und anderen Hindernissen (s/w) ● *Quadrat*: Bilderpuzzle. Ordnung im Chaos (s/w).

Versandbedingungen:

Um einen schnellen und problemlosen Versand zu gewährleisten, beachten Sie bitte folgende Punkte:

- Legen Sie pro bestellter Diskette DM 10,- als Scheck bei (bei Sonderdisk DM 15,-)
- Fügen Sie dem Betrag folgende Versandkosten (Porto + Verpackung) bei:
  Inland DM 5,– Ausland DM 10,–
- Legen Sie unbedingt einen Aufkleber mit Ihrer vollständigen Adresse bei.
- Für ausländische Besteller sei erwähnt, daß Sie mit Euroschecks auch in DM bezahlen können.
- Bitte legen Sie keine zusätzlichen Bestellungen bei (Abo, Zeitschriften, Platinen, etc.), da dies den Vorgang erheblich verlangsamt.
- Außerdem vermerken Sie bitte auf der uns zugesandten Bestellung „Public Domain ST".
- Zu dem Kostenbeitrag müssen wir leider pro Scheck eine Scheckgebühr von DM 0,50 berechnen. Verwenden Sie deshalb nur einen Scheck, auf dem die Gesamtsumme steht (z. B. für 2 Disketten = DM 25,50)
  für 4 Disketten = DM 45,50)

Auf Wunsch können Sie auch nach den alten Versandbedingungen bestellen, die wir zusätzlich aufrechterhalten.

Die Bezugsadresse lautet:

ST-Computer-Redaktion · Postfach 59 69 · 6236 Eschborn 1

Einführung in FORTH

Teil 5

Primitive oder nicht: Diese Frage stellt sich mit Sicherheit irgendwann jedem Forth-Programierer. FORTH-Worte können nämlich aus bereits bestehenden FORTH-Worten aufgebaut oder aber direkt in Maschinensprache definiert werden. Letztere werden als Primitive bezeichnet und können im volksFORTH mit Hilfe eines einfachen Inline-Assemblers erstellt werden. Lohnt es sich überhaupt, solche Maschinencoderoutinen einzubauen, oder könnte man nicht gleich ganz in Assembler programmieren? Die Antwort auf diese Fragen sowie die Definition von Primitiven, ihre Anwendung und der Umgang mit dem volksFORTH-Assembler wird in dieser Folge erläutert.

Warum Maschinensprache?

FORTH-Worte werden in der Regel aus bereits vorhandenen FORTH-Worten zusammengebaut. Die Ausführungsgeschwindigkeit eines solchen ohnehin schnellen „Highlevel"-Wortes kann in den meisten Fällen noch weiter gesteigert werden, wenn das betreffende Wort ganz oder teilweise in Maschinencode definiert wird. Solche Worte werden Primitive genannt, wobei lediglich der Umstand beschrieben wird, daß Primitive aus Maschinenbefehlen bestehen. Primitive werden z. B. für elementare Systemfunktionen, Betriebssystemaufrufe oder Fließkommaberechnungen eingesetzt. So besteht der Sprachkern der meisten FORTH-Systeme für den Atari ST bis zu ca. 75 % aus Primitiven. Den Rest machen Highlevel-Worte aus, die sich auf diesen Primitiven aufbauen. Allerdings bringen Primitive nicht nur Vorteile. So belegen Maschinencodedefintionen mehr Speicherplatz als ein Highlevelwort, in dem jedes Komponentenwort nur vier Bytes für seine Adresse belegt. Wesentlich schwerer wiegt jedoch der Nachteil, daß durch die Verwendung von Maschinencodedefinitionen die Übertragbarkeit von Quelltext zwischen verschiedenen FORTH-Systemen selbst auf der gleichen Hardware stark eingeschränkt wird.

Die Definition von Primitiven

Die Definition eines Primitiven wird durch das Definitionswort CODE eingeleitet. CODE wird in volksFORTH in der folgenden Form eingesetzt:

```
CODE <Name>
  :
  .
<Assembler Worte>
  :
NEXT
END-CODE
```

Wer die letzte Folge dieser Einführungsserie gelesen hat, wird sich bereits die Funktion des Definitionswortes CODE denken können. Seine Hauptaufgabe besteht darin, einen Wörterbucheintrag für den nachfolgenden Wortnamen <Name> aufzubauen. Dieser Wörterbucheintrag entspricht dem eines Highlevel-Wortes. Der einzige Unterschied ist der Inhalt des Codefeldes. Während der Zeiger im Codefeld eines Highlevel-Wortes auf eine Maschinenroutine zeigt, die für das jeweilige Definitionswort charakteristisch ist, enthält das Codefeld eines Primitiven einen Zeiger auf das eigene Parameterfeld. Genau da nämlich befindet sich die Maschinenroutine, die beim Aufruf des Primitiven zur Ausführung gebracht wird. Das Parameterfeld eines Primitiven wird durch die auf CODE folgenden Assembler-Worte gefüllt. Bei ihnen handelt es sich hauptsächlich um FORTH-Worte, deren Namen mit den 68000er-Mnemonics identisch sind, aber auch um Kontroll- und Steuerworte. Die Assembler-Worte befinden sich in einem separaten Vokabular mit dem Namen ASSEMBLER. Damit der Textinterpreter die Assembler-Worte bei ihrer Eingabe auch finden und ausführen kann, muß zunächst das Vokabular ASSEMBLER als „Context-Vokabular" vereinbart werden. Dies erledigt CODE durch Ausführung des Wortes ASSEMBLER. Falls Sie mit der Vokabularstruktur des Wörterbuches noch nicht vertraut sind, lesen Sie den Infokasten, der diesen Sachverhalt ein wenig ausführlicher erläutert.

Beendet wird die Definition eines Primitiven stets durch einen Sprung zu dem Adreßinterpreter NEXT (es gibt auch Ausnahmen, von denen später kurz die Rede sein wird). Dies wird durch das Assembler-Wort NEXT realisiert. NEXT kompiliert entweder einen Sprungbefehl nach NEXT oder, wie beim volksFORTH-Assembler, den Adreßinterpreter selber an das Ende des Parameterfeldes von <Name>. END-CODE beendet schließlich eine Maschinencodedefinition, indem es das ASSEMBLER-Vokabular aus der Liste der vom Textinterpreter zu durchsuchenden Vokabulare entfernt und das vor der Ausführung von CODE aktuelle Vokabular wieder als Context-Vokabular vereinbart.

Wozu ein Assembler in FORTH?

Der FORTH-Assembler ist ein Werkzeug, um Primitive definieren oder allgemein Maschinencoderoutinen in FORTH schreiben zu können. Mit einem Assembler wie diesem ist es allerdings nicht möglich, eigenständige Maschinenprogramme zu erstellen. Dazu müßte der Assembler um die Möglichkeit, eine lauffähige PRG-Datei erzeugen zu können, erweitert werden. Genausowenig ist es aufgrund der FORTH eigenen Schreibweise möglich, Quelltexte zu assemblieren, die für einen „richtigen" Assembler geschrieben wurden. In diesem Fall müßte man den FORTH-Assembler so erweitern, daß er auch in der Lage ist, bei Bedarf den Quelltext eines Motorola-Standardassemblers verarbeiten zu können. Das Erweitern des FORTH-Assemblers, der ja im Grunde kein homogenes Programm, sondern eine Aneinanderreihung an sich unabhängiger „Module" darstellt, ist relativ problemlos zu bewerkstelligen. Ob sich der Aufwand allerdings lohnt, sei einmal dahingestellt, möglich ist es jedenfalls.

Bei der Programmentwicklung wird der Assembler am Anfang nur selten eingesetzt werden. Der Grund dafür ist, daß Highlevel-Worte in der Regel schneller zu erstellen und leichter zu testen sind als Primitive. Erst wenn das Programm oder ein Programmteil fertig gestellt ist, kann man damit beginnen, einzelne Worte in Maschinencode zu recodieren. Da es dem Adreßinterpreter „egal" ist, welche Art von Wort er auszuführen hat (jedes Wort wird durch den Adressinterpreter auf die gleiche Art und Weise, nämlich

über seinen Codefeldzeiger, aufgerufen), kann ein Highlevel-Wort durch ein entsprechendes Primitive ersetzt werden, ohne den Rest des Programms ändern zu müssen.

Der volksFORTH-Assembler

Der Quellcode für den volksFORTH-83-Assembler befindet sich auf der zweiten der insgesamt zwei Disketten, die über den PD-Service bezogen werden können. Der Assembler ist sehr gut dokumentiert (die Dokumentation befindet sich in der Datei F_ASSEM.DOC im Ordner 1ST_WORD.DOC), so daß an dieser Stelle nur einige besonders wichtige Dinge besprochen werden sollen. Der Assembler kann z. B. durch die Befehlsfolge

USE ASSEMBLER.SCR <return>

1 LOAD <return>

oder einfach durch

INCLUDE ASSEMBLER.SCR

geladen und kompiliert werden. Wer sich vom Umfang des Assemblers überzeugen möchte, kann es einmal mit WORDS versuchen. Dazu muß aber das ASSEMBLER-Vokabular als Context-Vokabular vereinbart werden, was wie erwähnt durch Ausführung des Wortes ASSEMBLER geschieht. Nun steht dem Anwender der komplette Befehlssatz des 68000er zur Verfügung sowie zahlreiche Zusatzworte, mit denen z. B. Labels definiert oder Kontrollstrukturen (Entscheidungen und Schleifen) gebildet werden können. Definitionsworte für einfache Datenstrukturen wie z. B. 'EQU' oder 'DW' gibt es nicht, da man auch innerhalb einer Maschinencodedefinition sowohl auf alle Datenstrukturen des FORTH-Kerns (Variablen und Konstanten) als auch auf alle benutzerdefinierten Datenstrukturen zurückgreifen kann.

Um den Assembler besser kennenzulernen, sollen zwei Beispiele für typische Maschinencodedefinitionen vorgestellt werden. Dazu muß aber noch eine Besonderheit des FORTH-Assemblers vorangestellt werden: Auch innerhalb des Assemblers gilt die Umgekehrt Polnische Notation, d. h. es werden wieder zuerst die Operanden aufgeführt und dann erst der Befehl. So wird aus dem Befehl 'MOVE DO, D1' des Standard Assemblers in FORTH der Befehl ' D1 DO MOVE'. Nicht immer gelingt die Umsetzung so leicht wie in diesem Beispiel, so daß der FORTH-Assembler auch für erfahrerene Assembler-Programmierer recht gewöhnungsbedürftig ist.

Doch zurück zu den angekündigten Beispielen. Zunächst soll ein kleines Wort definiert werden, welches den Inhalt des obersten Stackelements mit zehn multipliziert. An diesem Beispiel sehen Sie, wie Maschinenroutinen über den Stack Parameter übergeben werden. Auch bei Maschinencodedefinitionen stellt der Stack die Schittstelle zur Parameterübergabe dar. Vorausgesetzt, der volksFORTH-Assembler wurde geladen, kann das Wort wie folgt aufgebaut werden:

```
CODE ZEHNMAL Ok
SP )+ DO MOVE Ok
10 # D1 MOVE Ok
D1 DO MULU Ok
DO SP -) MOVE Ok
NEXT Ok
END-CODE Ok
```

Zunächst erkennen Sie an dem stets erscheinenden 'Ok', daß sich FORTH während der Definition eines Primitiven im Ausführungsmodus befindet. Alle Worte des ASSEMBLERS-Vokabular werden direkt ausgeführt und kompilieren, wenn es sich um einen Maschinenbefehl handelt, den entsprechenden Prozessoropcode in das Wörterbuch. Es ist daher auch ohne weiteres möglich, während der Definition eines Primitiven z. B. irgendwelche Berechnungen durchzuführen, den Editor aufzurufen, eine Datei auszudrucken usw. Auch führt eine Fehlersituation nicht zu einem Abbruch der Definition, wie es etwa bei der Definition eines Highlevel-Wortes der Fall ist. Allerdings sollten Sie generell darauf achten, daß während der Definition einer Maschinencoderoutine der Inhalt des Stacks nicht verändert wird, da viele Assembler-Worte bestimmte Kontrollwerte zwischenzeitlich auf dem Stack ablegen, um die Richtigkeit von Adressierungsarten oder Kontrollstrukturen zu überprüfen.

Das Primitive ZEHNMAL wird durch das Definitionswort CODE eingeleitet, welches für den nachfolgenden Namen einen Wörterbucheintrag anlegt. Nun folgen die Assembler-Worte, die (da sich FORTH ja noch im Ausführungsmodus befindet) sofort ausgeführt werden. Da wäre zunächst der Befehl ' SP)+ DO MOVE', welcher das oberste Element auf dem Stack in das Register DO transportiert. Da dieses Element gleichzeitig auch vom Stack verschwinden soll, wird die indirekte Adressierung (SP steht für Stack Pointer bzw. Adreßregister A6, welches den Datenstackzeiger enthält) mit pre-inkrement verwendet. Der Multiplikator wird in Register D1 geladen. Es folgt eine Multiplikation des Inhalts von D1 mit dem Inhalt von DO durch den Befehl 'D1 DO MULU'. Anschließend wird der Inhalt von DO wieder, diesmal allerdings durch die Adressierungsart 'Indirekt mit Postdekrement' um eine Zahl in den Stack aufzunehmen, in das oberste Stackelement zurückübertragen. Das Assemblerwort NEXT kompiliert den Adreßinterpreter an das Ende des Wortes, damit nach der Ausführung von ZEHNMAL die Kontrolle wieder an das rufende Wort zurückgegeben werden kann. Die Definition des Primitiven ZEHNMAL, das sicher um einiges einfacher hätte definiert werden können, endet mit END-CODE, welches lediglich das ASSEMBLER-Vokabular „unsichtbar" macht und das vor dem Aufruf von ASSEMBLER als Context vereinbarte Vokabular wieder zum Context-Vokabular macht.

Wie bereits erwähnt, kann während der Definition eines Primitiven auch auf bereits existierende Datenstrukturen wie z. B. Variablen zurückgegriffen werden. Auch dazu ein kleines Beispiel:

```
VARIABLE ZAHL OK
O ZAHL !

CODE ZAHL=5 OK
5 # ZAHL R#) MOVE OK
ZAHL R#) D1 MOVE OK
NEXT
END-CODE
```

Die Ausführung von ZAHL=5 lädt schlicht und einfach die Variable ZAHL mit der Zahl 5. Mit ein wenig mehr Aufwand lassen sich so auch komplexere Datenstrukturen wie Listen oder Rekords bearbeiten.

Als zweites Beispiel soll gezeigt werden, wie sich z. B. GEMDOS-Aufrufe durchführen lassen. Zwar haben die meisten Entwickler von FORTH-Systemen diese Arbeit dem Anwender bereits abgenommen, indem sie fertige Worte zur Verfügung stellen, dennoch ist es recht lehrreich, eine solche Definition noch einmal nachzuvollziehen.

```
CODE MEDIACH? OK
. W (DRV R#) A7 -) MOVE OK
9 # A7 -) MOVE OK
$0D TRAP OK
4 A7 ADDQ OK
D0 SP -) MOVE OK
NEXT
END-CODE
```

Was bringt ein Assembler?

Wieviel man von einem Programm letzlich in Maschinencode definiert, bleibt dem Programmierer überlassen. Bestimmte Dinge, z. B. Betriebssystemaufrufe, lassen sich, sofern solche Worte nicht von vorneherein im System enthalten sind, nur in Maschinencode schreiben. Das gleiche gilt auch für Fließkommaroutinen, die ansonsten unerträglich langsam wären. Andererseits lohnt es sich kaum, Worte, die z. B. auf Tastatureingaben warten, in Maschinencode zu definieren. Als objektiver Maßstab läßt sich eigentlich nur ein Vergleich der Ausführungsgeschwindigkeiten heranziehen. Ein gutes Beispiel für einen solchen Vergleich bietet das Wort FILL, das einen bestimmten Speicherbereich mit einer Konstanten füllt. Abb. 1 zeigt FILL einmal in Maschinencode definiert und einmal in der entsprechenden Highlevel-Version. An den gemessenen Zeiten (ca. 2.8 Sec für die Maschinencoderoutine und ca. 1.05 Min für das Highlevel-Wort!) wird deutlich, wie groß der Geschwindigkeitsvorteil durch eine Maschinencodedefinition werden kann. Allerdings ist in dieser Gegenüberstellung der jeweilige Entwicklungsaufwand nicht berücksichtigt.

Nicht immer muß einer Maschinenroutine ein Wortkopf bestehend aus Namens- , Verbindungs- und Codefeld vorangehen. In manchen Fällen. reicht es, wenn sich eine Maschinencoderoutine irgendwo im Speicher befindet. Eine solche Maschinenroutine kann dann nicht über die Tastatur, sondern nur von anderen Primitiven über einen Sprungbefehl aufgerufen werden. Diese Maschinenroutine muß dann auch nicht mit einem Sprung zu NEXT enden, sondern kann z. B. mit einem 'RTS'-Befehl abgeschlossen werden. Um eine sog. „headerlose“ Routine zu erstellen, kann man wieder auf den eingebauten Assembler zurückgreifen. Da diesmal auf das Definitionswort CODE verzichtet wird, muß zunächst das ASSEMBLER-Vokabular durch Ausführen von ASSEMBLER „per Hand“ als CONTEXT-Vokabular vereinbart werden. Nun kann die Maschinencoderoutine mit Hilfe der Assembler-Worte auf die gleiche Art und Weise wie innerhalb einer CODE Definition erstellt werden. So lassen sich z. B. bestimmte Teile des Betriebssystems des Atari ST patchen, sofern sich das Betriebssystem im RAM befindet. In diesem Fall muß der Wörterbuchzeiger, der ja maßgeblich dafür verantwortlich ist, an welche Stelle ein Maschinenbefehl gespeichert wird, kurzzeitig auf die Adresse der zu ändernden Routine gesetzt werden.

Schließlich soll die Möglichkeit erwähnt werden, Maschinencoderoutinen mit Highlevel-Worten zu kombinieren. Durch das Wort ;C: wird das FORTH-System innerhalb einer Primitive-Definition in den Compilemodus geschaltet. Dies kann z. B. notwendig sein, wenn innerhalb einer Maschinenroutine ein Text ausgegeben werden soll:

```
CODE TEST
...
0< IF ;C: ." Fehler in der Eingabe"
ASSEMBLER THEN
...
```

;C: beendet eine Primitive-Definition und schaltet durch das Wort 'J' den Compilemodus ein (so einfach geht das in FORTH, probieren Sie es einmal aus). Da anschließend die Kontrolle wieder an den Adreßinterpreter übergeben wird, kann die Primitive-Definition nicht fortgeführt werden. Dies wäre nur möglich, wenn das aufgerufene Highlevel-Wort mit einem RTS-Befehl enden würde, wie dies z. B. bei FORTH-Systemen der Fall ist, deren Worte direkt verknüpft sind und daher keinen Adreßinterpreter benötigen.

Kompabilität zwischen FORTH-Assemblern

Wie bereits erwähnt, ist es mit der Übertragbarkeit von Assemblerroutinen zwischen verschiedenen FORTH-Systemen auf dem Atari ST nicht sehr weit her. So verfügt jede FORTH-Version für den ST, inzwischen sind es fast ein Dutzend, über ihren eigenen Assembler-Befehlssatz. Abhilfe schafft hier nur ein Blick ins Handbuch bzw. in den meist reichhaltigen Quellcode, der jedes FORTH-System begleitet.

Assemblierte Wortdefinitionen lassen sich mit Hilfe eines Disassembleres wieder in ihre ursprüngliche Form zurückübersetzen. Auch volksFORTH verfügt über einen Dissambler, der z. B. durch

INCLUDE DISASS.SCR

geladen werden kann. Der volksFORTH Disassembler kann zum einen zum Disassemblieren von Primitiven verwendet werden:

DISW <Name>

wobei <Name> der Name des Primitiven ist. Genausogut lassen sich aber auch allgemeine Maschinenroutinen bzw. Routinen des Betriebssystems disassemblieren. Der Disassembler muß dann in der Form

DIS <adr>

bzw.

LDIS <adr>

aufgerufen werden, wobei LDIS für Routinen ausserhalb von FORTH angewendet wird. Probieren Sie den Disassembler ruhig öfters aus. Sie lernen dadurch mehr über den Aufbau eines FORTH-Systems, als es im Rahmen einer Artikelserie oder auch eines Lehrbuches möglich ist. Überhaupt sollten sie einen Blick in den Quellcode von volksFORTH werfen bzw. diesen einmal ausführlicher studieren. Sie finden dort auch viele gute und anschauliche Beispiele für Maschinenspracheprogrammierung in FORTH.

Infokasten: Vokabulare in FORTH

Das FORTH-Wörterbuch ist in mehrere Untereinheiten aufgeteilt, die Vokabulare genannt werden. Die Unterteilung des Wörterbuches in Vokabulare bringt in erster Linie zwei Vorteile. Zum einen wird die Suche nach einem bestimmten Wort beschleunigt, wenn nicht das gesamte Wörterbuch, sondern nur ein kleiner Teil davon durchsucht werden muß. Zum anderen können funktionsverwandte Wörter in einem Vokabular zusammengefaßt werden. So werden beispielsweise alle Assembler-Worte in einem Vokabular mit dem Namen ASSEMBLER, alle Editor-Worte in einem Vokabular mit dem Namen EDITOR oder alle Fließkomma-Worte in einem Vokabular mit dem Namen FLOAT zusammengefaßt.

Vokabulare werden durch das Definitionswort VOCABULARY definiert. So kann z. B. ein neues Vokabular mit dem Namen NEU wie folgt definiert werden:

VOCABULARY NEU

VOCABULARY erzeugt für NEU einen Wörterbucheintrag, der aber ein wenig komplizierter aufgebaut ist als der Wörterbucheintrag einer normalen Wortdefinition. Damit das neue Vokabular auch vom Textinterpreter durchsucht wird, muß es als Context-Vokabular vereinbart werden. Dies geschieht einfach durch Eingabe des Vokabularnamens:

NEU ok

Die Eingabe eines Vokabularnamens erklärt das betreffende Vokabular zum Context-Vokabular, was bewirkt, daß dieses Vokabular bei einem Suchlauf mit FIND als erstes durchsucht wird. Nun wird das Vokabular zwar in einem Suchlauf berücksichtigt; damit aber auch Wörter in das neue Vokabular eingetragen werden können, muß das NEU auch als Current-Vokabular vereinbart werden. Dies geschieht durch :

NEU DEFINITIONS

Nach der Definition eines Wortes :

: TEST . " AHA " ;

wird dieses Wort nicht in das standardmäßig vereinbarte Vokabular FORTH, sondern in das Vokabular NEU eingetragen, das damit sein erstes Wort enthält.

In FORTH existiert eine ganz bestimmte Ordnung, nach der die einzelnen Vokabulare des Wörterbuches durchsucht werden. Diese Suchordnung kann mit ORDER ausgegeben werden.

ORDER

NEU FORTH ONLY NEU

Sie erkennen, daß das erste zu durchsuchende Vokabular NEU heißt. Dieses erste Vokabular in der Suchordnung wird als „Transient"-Vokabular bezeichnet. Transient bedeutet soviel wie flüchtig und beschreibt den Sachverhalt ganz gut, da dieses Vokabular nur solange in die Suchreihenfolge aufgenommen wird, wie kein anderes Vokabular als Context-Vokabular vereinbart wird.

Wird etwa durch

FORTH

das Vokabular FORTH erneut als Context-Vokabular vereinbart, so will FORTH nichts mehr von einem Wort mit dem Namen TEST wissen, da das Vokabular, in dem sich TEST befindet, nicht in die Suchreihenfolge aufgenommen ist:

TEST TEST haeh?

Allerdings werden alle neuen Wortdefinitionen nach wie vor in das Vokabular NEU aufgenommen, da dieses Vokabular immer noch als Current-Vokabular vereinbart ist. Erst durch

FORTH DEFINITIONS

wird der alte Zustand wiederhergestellt. Das gleiche kann auch durch das Wort ONLYFORTH erreicht werden, das generell den „Original"-Zustand wieder herstellt.

Das Transient-Vokabular kann durch das Wort ALSO permanent in die Suchreihenfolge aufgenommen werden. So wird durch

NEU ALSO

das Vokabular NEU wieder in die Suchreihenfolge mit aufgenommen und die Eingabe von

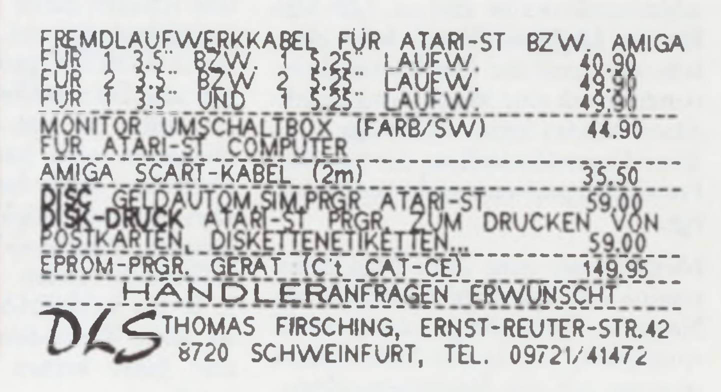
TEST AHA Ok

führt zu dem gewünschten Resultat. ALSO bewirkt lediglich, daß das momentane Transient-Vokabular auf dem Vokabular-Stack verdoppelt wird. Davon kann mit sich durch ORDER leicht überzeugen.

Ein anderer Aspekt, der sich durch die Vokabularstruktur des Wörterbuches ergibt, soll an dieser Stelle erwähnt werden. Durch das Aufteilen des Wörterbuches in einzelne Vokabulare kann ein Wortname zwei vollkommen unterschiedliche Bedeutungen haben, wenn er in zwei verschiedenen Vokabularen eingetragen ist. Durch ein einfaches Umschalten des Context-Vokabulars kann einer Gruppe von Worten eine vollkommen neue Bedeutung zugeordnet werden.

Abbildung 1

```
CODE FILL ( adr anzahl byte --- )
      SP )+ DO MOVE
      SP )+ D1 MOVE
      SP )+ D6 MOVE
      D6 REG) A0 LEA
      D1 TST 0< > IF
        1 D1 SUBQ
        D1 DO
             B DO AO )+ MOVE
           LOOP THEN
NEXT
END-CODE
: FILL1 ( adr anzahl byte --- )
   SWAP 0
         DO
           OVER 1 + >R
           DUP R> C!
         LOOP
         2 DROP
;
CREATE TESTFELD 10000 ALLOT
: BENCH 1
   100 0 DO
         TESTFELD 10000 65 FILL
       LOOP
;
: BENCH 2
    100 0 DO
          TESTFELD 10000 66 FILL
        LOOP
;
```

Leserbriefe



Betr.: Probleme mit fehlerhaften Desk-Accessories auf der Festplatte. Folgendes war mir passiert:
Ich hatte ein fehlerhaftes Desk-Accessory auf Partition C meiner Festplatte gespeichert. Beim anschließenden Kaltstart mit Festplattentreiber, bei dem bis zu 6 Accessories geladen und installiert werden, löste das Laden des fehlerhaften Accessorys stets einen Reset mit erneutem Booten aus!
Aus dieser endlosen Schleife gibt es scheinbar keinen Ausweg. (Auch der Druck auf die RESET-Taste führt nur zu erneutem Booten). Daß mir schließlich doch eine Lösung einfiel, verdanke ich der Tatsache, daß ich zu den relativ frühen Käufern des ATARI zähle, die noch im Besitz der Disketten-TOS-Version sind. Bekanntlich ist für den Betrieb der Festplatte das TOS im ROM erforderlich; die Disketten-Version arbeitet unzuverlässig, erkennt nur eine Partition, beschränkt stark die Zahl der Ordner sowie deren Schachtelungtiefe und (hier liegt der Schlüssel zur Lösung:) Desk-Accessories werden nicht automatisch beim Kaltstart von der Festplatte geladen.
Damit ergibt sich folgender Lösungsweg:
Man speichert (bei abgeschalteter Festplatte) die Disketten-TOS-Datei TOS.IMG auf die Festplatten-Boot-Diskette und führt einen Kaltstart durch (Festplatte und Rechner). Die Festplatte erscheint als Laufwerk C. Nun löscht man das fehlerhafte Accessory sowie auf der Bootdiskette das nun nicht mehr erwünschte TOS.IMG. Nach Ausschalten von Rechner und Festplatte sowie erneutem Kaltstart ist der alte Zustand wiederhergestellt. Das gleiche Problem ergibt sich übrigens, wenn zu einem Desk-Accessory die zugehörige RSC-Datei fehlt.

Stefan Bauer, Reutlingen

Mit einigem Bedauern, muß ich, der ich die Beiträge der „ST" seit Anfang an verfolge, feststellen, daß Sie im großen ganzen über die Sichtweise einer Hauspostille der Atari Corp. nicht herausgekommen sind.
So ließen leider die Interviews, die Sie mit leitenden Herren dieser Firma führten, jeden journalistischen Biß vermissen. Kein Wort z. B. über die Sitte dieser Firma, die tollsten Sachen vollmundig anzukündigen um sie dann doch sang- und klanglos untergehen zu lassen (s. MS-DOS-Emulator, das Trauerspiel währt nun über ein Jahr!), oder die vergleichsweise schlechte Verarbeitungsqualität.
Ähnliches muß leider auch über die Tests von Anwender-Software gesagt werden. Bei Ihnen scheint es schlechte Software überhaupt nicht zu geben. Sie sind über jedes kleine Feature so begeistert, daß Sie anschließend vergessen, was auf dem Markt der Kompatiblen schon an Software vorhanden ist. Und genau mit diesem Markt sollte sich der ST messen lassen, sonst ist er eben doch nur ein aufgemotzter Home-Computer.
Ansonsten gefällt mir Ihre Zeitung gut, insbesondere die Sprachkurse, ich würde mir jedoch wünschen, daß so ein Leserbrief wie der von H. Vasilescu (4/87) nicht das einzige Kritische bleibt, was in der ST zu lesen ist. Geeignete Themen gibt es ja bei Atari in Hülle und Fülle (s. o.), ich glaube, sie müßten nicht unbedingt lange auf die Suche gehen.

Stephan List, Uetersen

Die Anzeige in ST 6/87 mit dem Werbespruch „Programme, die auch Frauen bedienen können" suggeriert bei Frauen Beschränktheit im Umgang mit dem Computer. Die Anzeige ist dumm und frauenfeindlich.
Als regelmäßige ST-Leserin und professionelle Computerbenutzerin bin ich empört darüber, daß Ihre Zeitschrift Platz für solche Darstellungen von Ignoranz einräumt. Das kann ja weder im Interesse Ihrer Leser und Leserinnen noch Ihrem Ruf dienlich sein, Frauenfeindlichkeit zu inszenieren.

Vivien Marx, Frankfurt

In der Juni-Ausgabe der von Ihnen herausgegebenen ST-Computerzeitschrift wurde eine Anzeige mit dem Programm, die auch Frauen bedienen können veröffentlicht.
Diese Anzeige trägt meiner Ansicht nach nicht gerade dazu bei, als Frau sich vermehrt mit Computern zu beschäftigen.
Solange das Geschlecht als Kriterium für Anwenderfreundlichkeit verwendet werden kann, wird sich auch daran nicht viel ändern. Solche Anzeigen sind gegenüber allen weiblichen Lesern ihrer Zeitschrift eine schlichte Unverschämtheit.

Ute Schick, Mannheim

Ich bin reichlich enttäuscht und verärgert darüber, für wie frauenfeindlich die Leser/innen Ihrer Zeitschrift gehalten werden. Eine Anzeige mit der Titel-Zeile: „Programme, die auch Frauen bedienen können" zeigt ja allzu offensichtlich, daß das Lesepublikum als recht überheblich (soweit es männlich ist) bzw. als recht dümmlich (soweit es weiblich ist) eingeschätzt wird.
Ich bin nicht bereit, mir solche frauenfeindlichen Ansichten unterstellen zu lassen.

Lorenz Hölscher, Aachen

Die Redaktion schließt sich den vorgebrachten Meinungen an und bittet die Veröffentlichung dieser Werbung zu entschuldigen.

In ST-Computer vom Juni 1986 veröffentlichten Sie einen kritischen Leserbrief zum Textverarbeitungsprogramm 1st Word Plus. Diese Kritik möchte ich vertiefen:
Jede (auch noch so billige Schreibmaschine) ist in der Lage, einen Text mit einem Zeilenabstand von 1 1/2 zu schreiben. Nicht so 1st Word Plus! Dieses Programm, das in vielen Aspekten (z. B. Fußnotenverwaltung) den anspruchsvolleren Anwender interessieren dürfte, arbeitet lediglich mit ganzzeiligen Abständen. Eine gewisse Abhilfe schafft zwar ein von Atari zur Verfügung gestellter Druckertreiber; er erlaubt es aber nicht, verschiedene Zeilenlineale (z. B. die für wissenschaftliche Texte sehr wichtige Kombination von 1 1/2zeiligem Textlineal und einzeiligem Fußnotenlineal) sinnvoll einzusetzen, da das Programm die unterschiedlichen Zeilenabstandbefehle des Druckertreibers nicht in seine Seitenformatberechnungen einbeziehen kann. Zudem ist dieses Verfahren sehr umständlich und zeitaufwendig, was einer Textverarbeitung nicht würdig ist! Ein weiterer Fehler des Programms besteht darin, daß nach dem Formatieren eines Textes ein ca. zweizeiliger Abstand zwischen die einzelnen Fußnoten gesetzt wird. Dies widerspricht ganz klar den Angaben im Handbuch auf S. 34. Noch schlimmer wird es natürlich, wenn man den 1 1/2-zeiligen Druckertreiber benutzt: der Abstand vergrößert sich entsprechend.
In einem Brief an Atari habe ich diese Probleme geschildert. Die Reaktion war beschämend und hat mein bisher gutes Bild von der Firma nachhaltig gestört: sie bestand darin, eine Diskette mit Druckertreiber zu schicken, „in der Hoffnung mir damit gedient zu haben". Auf die Vorschläge, die ich zur Abhilfe gemacht habe, ist man mit keinem Wort eingegangen. Von meinem Händler habe ich dann erfahren, daß nach Auskunft des Atari-Bezirksvertreters Atari nicht daran interessiert sei, angesichts des Verkaufspreises noch Geld in das Programm zu investieren.
1st Word Plus ist an sich ein gut konzipiertes Programm und für einen Anfänger leicht erlernbar; aber angesichts der Schwächen kann man es zur Zeit für einen ernsthaften Anwender nicht empfehlen.

Kai-Uwe Kock, Münster

Leserbrief ST 2/87
Dort wird bemängelt, daß zu viele Pascal- und C-Listings in Ihrer Zeitschrift vorkommen. Er vermißt zudem noch viele Software-Tests.
Ich bin der Meinung, daß ein Computer wie der Atari nicht zu einem reinen Software-Werkzeug degradiert werden sollte. Wer hat denn überhaupt das Geld, um all die Software zu kaufen? Die meisten Atari-Besitzer wohl kaum. Die von Ihnen aufgeführten Listings sind sehr gut und enthalten viele Tips. Auch wenn man die Programme nicht einfach blind eintippt und laufen läßt, kann man durch ein kleines Studium des Programms wertvolle Anregungen gewinnen. Ein Computer ist doch sicherlich auch dazu da, programmiert zu werden. Jeder, der ernsthaft arbeitet, hat immer ein spezielles Problem. Es gibt meines Wissens keine Software, die alles kann. So, wie es Autofahrer gibt, die die Zündkerzen nicht selber auswechseln (auch wenn es erheblich billiger sein sollte), so gibt es scheinbar noch Menschen, die sich überhaupt nicht um das kümmern, was im Rechner läuft. Sie lassen sich alles vorsetzen und konsumieren nur noch: Denken ist unerwünscht!!!

Gerald Prior, Bielefeld

Bei der Arbeit mit dem XLISP 1.5-Interpreter bin ich auf einige Unverständlichkeiten gestoßen. Es handelt sich dabei um die Funktionen „remove" und „member": Im Gegensatz zu COMMON-LISP, daß dem Buch von Winston-Horn zugrunde liegt, haben die beiden Befehle in XLISP noch zwei Parameter mehr, nämlich „key" und „test". Im manual zu XLISP heißt es dazu:
key the keyword :test or :test-not
test the testfunction (default to eql).
Vielleicht können sie mir sagen, was der XLISP-Interpreter an diesen beiden betreffenden Stellen haben möchte, so daß er die Befehle so ausführt, wie sie eigentlich ausgeführt werden sollten.

Kay Sendelbach, 5000 Köln

KEY ist ein optionales Schlüsselwort. Wenn Sie es benutzen wollen, geben Sie entweder :test oder :test-not ein. Dann müssen Sie aber als nächsten Parameter eine Funktion eingeben z. B.: (lambda (a b) (not (eql a b))). Wenn Sie das Schlüsselwort :test verwendet haben, gibt remove eine Liste zurück, in der alle Elemente fehlen die die Testfunktion erfüllen. Haben Sie das Schlüsselwort :test-not verwendet, erhalten Sie alle Elemente der Liste die die Testfunktion nicht erfüllen.

Bei mir bestehen folgende Probleme:
1.) Bei einigen meiner Disketten ist es mir manchmal nicht möglich, einen Ordner beim ersten Doppelklick zu öffnen, auch bei einmaligem Anklicken des Ordners und Hineinschauen in das Disketteninfo scheint der Ordner leer zu sein. Zwar beginnt das Laufwerk zu laufen, jedoch wird das Directory der Disk scheinbar noch einmal geladen und wieder angezeigt; das Fenster flimmert dabei so, als ob ein Ordner geöffnet worden wäre. Erst jetzt ist es möglich, an den Inhalt des Ordners zu gelangen. Die meisten meiner Disketten habe ich mit Superformat doppelseitig formatiert. Dieses Problem ist dann besonders kritisch, wenn ich z. B. aus dem Basic-Programm heraus an ein Programm möchte, das in einem Ordner liegt. Es ist dann unmöglich, dieses zu erhalten. Ich muß dann das Programm unterbrechen und das gewünschte Programm aus dem Ordner heraus kopieren, um es aus dem Basic-Programm zu erreichen.
2.) Ich besitze eine Brother CE-500-Schreibmaschine, die ich an den Rechner angeschlossen habe. Bis jetzt ist es mir nicht gelungen, den Ausdruck mit 1st Word 1 1/2 oder 2zeilig auszugeben.
Ich hoffe, daß mir jemand bei meinem kleinen Problemen helfen kann.

Leander Pillmann, Bretten 2

Ein häufiges Druckproblem ist die Einhaltung der Seitenlänge bei 1 1/2-zeiliger Schreibweise, soweit dies programmseitig nicht unterstützt wird. Die Umschaltung auf 1 1/2 Zeilen erfolgt bei Epson-Druckern im ASCII-Modi mit 027/051/054. Damit das Papier nicht vom Textprogramm her auf das programmseitige (falsche) Textende transportiert wird, geben Sie nach dem Textende, spätestens nach der 43. Zeile ein: (Asci-Modi) 027/051/001 (Tip: als Prozedur oder auf Funktionstasten abrufbar speichern).
Damit Sie ohne Beachtung des Papiertransports des Druckers mehrere Dokumente hintereinander an den Drucker geben können, sind noch in den Druckparametern des Textprogramms einzugeben:
Initialisierung: 027/051/036 = 1-zeilig oder 027/051/054 1 1/2 zeiliger Abstand sowie als
Reset: 012 = Transport auf das Seitenende.
Nun kann endlich auch ein Druckspooler genutzt werden.

Manfred Hagedorn, Hagen

An alle VIP (Lotus-123) Profi's!
Wer kann mir bei folgendem Problem helfen: Ich möchte eine Tabelle über ein Menü und eine Dateneingabemaske mit VIP eingeben. Um nun die Daten von der Eingabemaske in die Tabelle zu schaufeln, gibt es im Lotus-123 den Befehl /rtxx.yy (dabei sind xx und yy Tabellenkoordinaten des Empfangsfeldes), oder zu englisch: Range Transpose.
In VIP gibts diesen Befehl nicht, der Transpose-Befehl ist nicht einmal im VIP-Menü selber enthalten! Wie kann ich diesen Mangel umgehen? Ich kenne keinen anderen (akzeptablen) Weg, den Maskeninhalt an eine Tabellenposition zu kopieren, zumal die Maske vertikal und die Tabelle horizontal aufgebaut ist.
Wie kann ich den Ablauf eines Macros beschleunigen (z. B. Ausblenden der Befehlsfolgen)? Hat mir jemand gar ein kleines Beispielprogramm? Ich bin um jeden Hinweis sehr dankbar.

Rene Lieb, Cham (Schweiz)

Sehr geehrte Damen und Herren
Ich besitze einen 520ST+ mit 2 Laufwerke und möchte Daten mit einem Schneider Joyce austauschen (direkter Weg mit Kabel). Wer hat diesbezüglich Erfahrungen gesammelt?

Josef Kirchhartz, Neuss

Ich bin vor kurzem vom C64 auf den 1040 ST umgestiegen und benutze heute beide Computer von Fall zu Fall. Ich möchte nun gerne vorhandene Bilder aus dem Printfox-Format des C64 in das Doodle-Format des Atari überspielen (und umgekehrt). Wer kann mir Hinweise hierzu geben, welche Hard- bzw. Software hierfür erforderlich ist? Vielleicht hat ja schon jemand eine Lösung dieses Problems!

Martina Nowak, Haan 1

Bitte teilen Sie mir mit, ob Ihnen eine Möglichkeit (Software) bekannt ist, eine Hardcopy mittels eines OKI-ML 192 zu erstellen.

Dr. L. Pape, Nieheim/Westf.

Seit Februar 87 habe ich 1st Word Plus mehr oder weniger in Betrieb. Bis heute ist es mir nicht gelungen, meine zwei Drucker optimal mit 1st Word Plus einzusetzen. Nun hoffe ich, daß mir ein Leser mit gleichem Drucker helfen kann.
Gerätekonfiguration:
- Atari 1040ST mit Monochrom-Monitor
- Olivetti Praxis 40 (am Drukker-Port)
- Epson MX 80F/T mit Rom-Satz MX80 Typ 3 (am Modem-Port)
- 1st Word Plus 1.89 (BRD)

Problem 1: Olivetti Praxis 40
Beim Ausdruck mancher Texte (Briefe), werden die Leerzeichen nicht berücksichtigt. Text ohne Leerzeichen. Jedoch mit dem Drucker-Treiber „STANDART. CFG" wird der Text bis auf die Umlaute richtig formatiert ausgegeben. Leider fehlt auf der 1st Word-Diskette das entsprechende HEX-File.
Die Funktion „Unterstreichen" kann nicht aktiviert werden.
Problem 2: Epson MX 80F/T
Beim Grafikausdruck (z. B. TEST auf 1st Word-Diskette), werden nicht alle übertragenen Bytes, als Bitmuster, sondern als ASCII-Zeichen ausgedruckt.
Nach dem Einlesen eines Grafikbildes in einen 1st Word-Text, wird der Ausschnittrahmen nach rechts größer angezeigt, als dieser Ausschnitt in Wirklichkeit ist.
Schmal-, Normal- und Breitschrift sind nicht in einer Zeile kombinierbar.
Problem 3: Drucken über seriellen Port
Kann man 1st Word Plus, so abändern, daß nach dem Laden sofort der Drucker am seriellen Port initialisiert ist.
Zu erwähnen bleibt nur noch, daß beide Drucker an anderen Systemen, wie z. B. Commodore C64, einwandfrei funktioniert.

Klaus R. Obert, Lahr

Unser Farbfernseher Grundig Eleganz 4215 hat leider keine Scartbuchse. Gibt es eine Möglichkeit, selbst einen Scartanschluß in die „Glotze" einzubauen?
Vielleicht hat dies einer der Leser schon an obigem oder einem anderen gängigen Modell erfolgreich durchgeführt.

Ulrich Müllerschön
7000 Stuttgart 80

Wie installiere ich am besten meinen Lattice-C-Compiler mit Thempus-Editor auf einen Atari-ST mit 1 MByte-Ram? Wie groß sollte die RAM-Disk gewählt werden und was sollte sie beinhalten, um auch größere Programme compilieren zu können? Für ihre Mühen danke ich.

Arthur Milbrandt
8000 München 40

Wie alle GFA-Anwender wissen, ermöglicht der Befehl FORM INPUT...AS(String) die Ausgabe des Strings, welcher sodann mittels „Delete" oder „Backspace" gelöscht, bzw. abgeändert werden kann. Es ist eben nicht sehr anwenderfreundlich, daß dieser Befehl nicht im Überschreibmodus betrieben werden kann; deshalb frage ich, ob Sie eine Lösung für obiges Problem kennen. Weiteres würde ich gerne wissen, ob und wie es möglich ist eine Taste so zu belegen, daß bei deren Betätigung (an beliebiger Stelle im Programm) ein bestimmtes Kommando ausgeführt wird. Ich denke dabei daran, daß ja auch die Tastenkombination Control. + Alt + Shift die Programmausführung stopt.

Hannes Laufenthaler
GAIS/Südtirol BZ

Kleinanzeigen

BIETE HARDWARE

Comp. + Zubeh. sehr preisgünstig Info 05732/7743 ab 16.30 Uhr

★ ★ SUPER Atari 520ST + ★ ★ 1MB, TOS im ROM, Weide Echtzeituhr, Disk 314, Monitor SM124, Maus Software, Wg. Mega ST DM 1.300,- 04181/35767 n. 19 Uhr

520ST + 1MB ROM-TOS, 2xNEC 1035LW Monitor SM124, komplett einwandf. Handbuch + Logo + Basic VB 2300 DM. W. Feller, Josef Feller-Weg 6, 8882 Lauingen

DA-AD-Karte f. Rho-Bus-System 1/2 Jahr alt 750,– 0205484962

Blittertos und PRG's brennen wir auf Eprom • T. 05171/3723

SF354, ROM-TOS 07031/275295

520ST +, 1MB, Romtos, SF354, SF314, SM124, Software, Bücher VB 1500,– DM 02372/51155 ab 19 Uhr 02352/71201

260ST (1MB, Uhr) + Monitor SM124 + Floppy SF354 + Drucker DMP3000 + Maus + Anwend.-PRG + Spiele + Bücher usw. 1898,– DM, Tel. 02241/334923

SF354 DM 150,- Tel. 0241/520812

ST 260 + dtsch., 1MB, Fast-TOS-ROMs + Floppy 720KB + Maus, neu m. Gar., DM 999, T. 06405-3350

ATARI-Echtzeituhr akkugepuffert Einbau ohne Löten DM 98 Romport nicht belegt T.02337/1239

SF354 150 DM D. Braun 06051/66472 18 – 20 Uhr

NEC-FD1036, 245 DM Tel. 05446/1242

Verkaufe FET-Farbdigitizer, neu für DM 369,–, Tel: 06484/6002

A5 Sound Sampler (mit Softw.) Preis 160,– DM Tel. 0711/560937

SM 124 Preis VB 0211-717141

ALADIN + PD-MAC Programme Österreich: 05244/37223

Atari 520 ST + incl. Maus, SF314 u. SM 124, alles neuw. für VB 2000,– DM zu verk. Tel: 06082-1062 oder 2238

EPSON LQ 800, 24N-Drucker, neuwertig, VB 1180, 0931/26321

520 ST +, 1040 KB, TOS im ROM 2 SF 354 Laufwerke, CSF-Gehäuse SM 124 Bildschirm, ST-Clock mit Software .... DM 1.580,– Telefon .... 089 / 310 88 72

ROM-TOS m. Schnell-Lader in U7 100 DM (VB), Tel. 02630/7525

Thomson Neochrome-Monitor 3/4 J. alt NP 400, VB 250 Tel. 07151/81456 (Steffen)

Verk. Atari Farbmonitor SC1224 DM 500,- Tel. 06236/54806 n. 18⁰⁰

Farbdrucker Epson Jx80 1050 DM Farbplotter Epson HI80 900 DM Drucker OKI ML80 DM 150,– Tel. 08139/6444

Verk. Atari ST Floppy SF 354 für 140 DM, 7 Monate alt. Adresse: Markus Theimer, Schwedenstr. 2, 6203 Hochheim

5 1/4" Laufwerk f. ST Neu 450,– Drucker TA 200,– 02861/62628

Verk. neuw. SF354 T. 0911/441154

★ Floppy-Stationen Atari ST ★
★ Einzelstation 349 DM ★
★ Doppelstation 659 DM ★
★ 5.25" 40/80 Tr. 498 DM ★
★ ★ 100% ST-Comp., Netzteil ★
★ ★ eingeb., mit NEC 1036A ★
★ ★ kompl. anschlußfertig ★
★ – NEC FD 1036A: 225 DM ★
★ – SF354 Umbau mit 1036A ★
★ – nur 288 DM! – ★
★ FISCHER Computersysteme ★
★ Tel: 06151/51395 ★

★ Atari ST Diskettenstation ★
Jetzt auch als Erstlaufwerk
1x3,5" 726KB Abm. 105x30x220mm (NEC) eing. Netzteil nur 369,–
2x3,5" 1,4 MB nur 679,–
NEC FD 1036 A für ST nur 238,–
★ Weiteres Zubehör günstig ★
★ DELO Computertechnik ★
★ Tel: 0231/339731 o. 331148 ★

■ Echtzeit-Hardwareuhr 98,–
■ Eprom-Bank 128 K (1040) 45,–
■ Leerplatine –"– 25,–
■ Ing.-Büro für Datentechnik
■ Glätzlstr. 10, 8070 Ingolstadt
■ Tel. 0841/57272

Gute 2,50 DM-Disks:0431/569216

2 MB-Speichererweiterung 950,- auf 1 MB 220 DM; Eprommer 165,- Kersting + Rehrmann Sperberweg 10, 3500 Kassel Tel. 0561-777109, 61075, 283522

EPROMMER mit Textool 145 DM Accu-Uhr nur 35 DM ☎040-862771

★ NEC-1036 Diskstationen ★
★ Einzellaufwerk 395,– ★
★ Doppellaufwerk 695,– ★
★ anschlußfertig, superflach ★
★ Bauw. Betr.-kontroll. ★
★ ATARIPOINT 07031/52607 ★

Super Disks: 2,15 DM:0481/72839

ATARI-Echtzeituhr akkugepuffert Einbau ohne Löten DM 89 Romport nicht belegt Tel. 02337/1239

★ Wegen Systemaufgabe ★ Verk. ATARI 1040STF + Maus + Trackb + SF314, Col. Mon. Sanjo, ST-Pascal, div. Software (Original) Rolf Hitz, Dummersdorfer-Str. 97 24 Lübeck 14, zus. 2500 (kein Tel.!)

DTP Job-Unit 3 Mon. 0228/674928 Laserdr. Scanner etc. VB 17900

SF 354, 1J, 150 DM 089/6906528

BIETE SOFTWARE

Von Lehrer für Lehrer! Notenverw.prg. Markbook plus in GEM. Über 20 versch. Funkt. f. alle gäng. Notensyst. Spez. vers. auf Wunsch; Info gegen Freiumschlag. H. J. Merkel/Nahestr. 28/6600 SBR

SOFTWARE sehr preiswert. Rufen Sie an 05732/7743 ab 16.30 Uhr

PD-SOFTWARE, die noch keiner hat. Alle Prg frei auswählbar! Und das Beste: NUR 1 1/2 PF pro KBYTE!!!! Umfangreicher & KOSTENLOSER Katalog! Arne Zingel, Vermehrenring 11a, 24 Lübeck 1

BS Handel, Vers. 1.20, Originalverpackt, ungebraucht, umständehalber für DM 460,– zu verkaufen. Tel. 07232-70708

50 Anleitungen für Anwendungen u. Spiele + div. Originalprogr. wg. Systemwechs. 02238/51394 o. 52795

PD: Einzelkopien von Prg. Ihrer Wahl!! D. Metz, Dyckburgstr. 18 4400 Münster

Wegen Systemwechsel verk. ich meine Originalsoftware f. ST. Liste gegen Rückporto (1,30) Armin Panzer, Sonnenstr. 1 8700 Würzburg

★ Hier geht's rund. ST-User ★ PD-Service (1+25), An-/Verkauf aktuell · günstig · schnell · neu! Gratis-Info: Carsten & Marcus Postfach 65 06 02, 2 Hamburg 65

Scanner für Atari ST steckbar Anschluß über Joystickport mit Anschlußkabel und Software Einbauanleitung DM 285,70 incl. MwSt Technisches Büro Knäbel 82 Rosenheim, Ebersbergerstr. 48, 08031/83717 Händleranfragen erwünscht.

DEPOTVERWALTUNG-ST Prg. Übersichtlich, versch. Währungen Diskette DM 49,–. Bei IVOREK Postfach 1301, 6204 Taunusstein 1

Atari ST div. Originalprogramme Tel. 05171/3723

Schweiz: Publ. Domain-Software PD01-PD51 aus ST-Comp-Zeitschr. Zwei Disc auf einer Markendisc z.B: (1+2) (3+4) (5+6) (15+16) etc. sFr. 10 pro Disc, Check od. Bar PD-Soft, Postfach 8, 8602 Wangen

MIDI Sequ-PRG C-Lab "Creator" f. 490 DM statt 600 DM · CMX Computermischpult (Info anford.) 030-8559848

Verkaufe Orig. Hacker II u. Skyfox für ST VB je 55 DM 0711/796107

Adimens und Beckertext – beide Original. Tel. 0511-874209

VIP (150,-), Textomat (25,-), Printmaster (25,-), Starglider (35,-), Flight Simulat. (85,-), Steuer Tax 2.6 (50,-) Tel.: 06121-502392

ST-SOFT? Tel.: 0212/53387

Disk-Etiketten bedrucken (ATARI St) Anf. Einlesen v. Files u. manuelle Erstellung unter GEM. Ermittelt File-Länge u. freie Kapazität d. Disk. Dipl.-Ing. R. Zepernick, Leipziger-Str. 69, 2120 Lüneburg. Preis 45,– DM einschl. Disk.

Lohnsteuerjahresausgleich/Einkommensteuer 86/87. Grunddaten + Ergebnis speichern. Was Wäre Wenn Lohn- u. ESt-Tabellen 1986/87 30 DM Vork. J. Höfer, Grunewald 2a, 5272 Wipperfürth 02267/3368

ST-PDS Public Domain Service Über 160 Disketten f. ATARI ST Katalog anfordern bei ST-PDS 1 Berlin 65, Soldinerstr. 4 NEU! Auch Hard- u. Software verschiedener Hersteller! 030/4948820

Original BS-HANDEL Preiswert abzugeben. Thomson Farbmonitor 36382 ebenfalls zu verkaufen. Nach 19.00 Uhr 06121/520644

Baustatik / Ausschreibung Holz – Beton – Stahl – Wärme Angebot – Massen – Preissp. Programmservice Schmidt Burgstr. 4 · 5376 Marmagen Tel.: 02486/7384 (7417)

Public-Domain Kopierservice über 200 PD-Disk ★ Gratiskatalog Interessante Preise ★ einseitig 5,50 DM ★ zweiseitig 6,50 DM M. Schönfelder · Tel.: 02954/1050

Verkaufe Originalprogramme: K-Spread DM 95, Monostar DM 65, Textomat DM 65 05207/4523

Js Gem Da, GTI-Datenbank: 110,– Profipainter: 45,– u.a. Orig. S.W. Suche: SF 314 Tel. 02266/7894

75.000.000 Bytes PD-Software Spiele, Anwendungen aller ARt, Grafik- und Musikprg's, viele Grafikdemos, Sie werden überrascht sein. Ausführliche Beschreibungen über unser PD-Angebot mit weit über 1000 Programmen auf 250 Disketten gibt es bei:
Gerald Köhler
„PD-Service"
Mühlgasse 6
6991 Igersheim
Tel. 07931/44661 (24h-Service)

Verk. Wordstar 3.0 mit Maile Merge für Atari ST für 120 DM J. Bauer, Tel. 06441/48191

■ Entwicklung v. Individual
■ Software unter GEM!!
■ CS-Share Aktiendepotverwalt.
■ INFO: 07131/72639 oder 81163

Wärmebed. DIN 4701 + K-Zahl DM 110; Rohrnetzber. DM 60; Demodisk DM 10 nur Vorkasse von J. Binder, Eichendorffstr. 15, 5030 Hürth

Grafik-Bilder von Degas, Neo, u.a. 70 Disks, Katalog von M. Frey Rheinstr. 12A, 6538 Münster-Sarmsheim

! Bundesliga auf dem ATARI ST Hausbedienung, Analyse, Saison-Editor, nur 29,– Disk. 0291/1785

Wieviel min noch auf Band? Zählwerk-Echtzeit-Umrechner!!! Für alle Kass- und Rek-Typen!!! Bestellungen an: Willsoft Am Trieb 10, 8763 Klingenberg 2 ★ 20 DM ★ Info gratis

Verk. orig. ESML-Programmbibl. für Simulation & Datenanalyse (C Source), 49 DM. T. 07557/8639

Stabwerk- u.v.a Statik-Progr. sowie über 60 Mathem.-Progr. Info (1 DM) bei: Dipl. J. Bullmann, Zur Hindenburgschleuse 3 3000 Hannover 71

GfA-DRAFT 190,– 0221/314039

ATARI Public Domain Service
T Disk aus der ganzen Welt
A Rainer Langner
R Telefon: 0201/29 20 00
I Brandhoffs-Delle 11, 43 Essen

Megamax-C m. dt. Anleitung 440,– ST-Pascal + 200 DM 1ST-Word 160 DM oder Tausch gegen andere Programmiersprachen (Apl, Modula, Lisp, Prolog) 07463/1771 ab 17^00

■ SPORTTABELLE – universelles Tab. prg. unter Gem, 25,–, Demo 5,–, Heinze, Burgallee 27, 6450 HU

Schweiz: Publ. Domain Software PD01 – D51 aus ST-Comp-Zeitschr. Zwei Disc auf einer Markendisc z.B: (1+2) (3+4) (5+6) (15+16) etc. sFr. 10 pro Disc, Check oder Bar PD-Soft, Postfach 8, 8602 Wangen

Orig. Lattice C, Bücher: „Von Basic zu C" und „Kernighan: Programmieren in C", alles zusammen für 200 DM. Werner, Am Steig 2, 3508 Melsungen

Wärmebedarfs./KZahlberechnung DIN4701/83 GEM Bedienerführung Dipl.-Ing. V. Koch, Am Mehnacker 11 3563 Dautphetal 3, Tel. 064687652

★ ★ DER ABSOLUTE HIT! ★ ★ Formatier-Accessory auch für das unmöglichste Format. Mit 35 DM sind Sie dabei. Bestellung bei FRANK LAYER Leonorenstr. 49, 7000 Stuttgart-70

1st-Word+; Side Click; DB-Master 1; VB; Tel. 02841/503207

Journal Buchführung m. Atari St Für betriebl./priv. Nutzung: Max. 39 Konten, 4800 Bewegungen Ust/Vst, Abschluß, Etatplanung DM 59, Kitta, 02293/3338, n. 18^00

■ ★ PD zum Selbstkostenpreis ★ ■
■ Pro Disk aus ST-Computer 6,–
■ IO = DM 50,- plus 5,– Versand
■ Bar/Scheck. Weitere Sonderangebote. Liste mit anfordern! Jörg, Prilmayerstr. 26, 8058 Erding

ATARI ST – LOTTOSTATISTIK für 39 DM, 20 seit. Dokumentat. vorab gegen 2,40 in Briefmarken F. Klein, Dessauer Str. 5a 89 Augsburg 21, T. 0821/812157

PD-Software
Freie Zusammenstellung der Prg Bei freier Wahl: DM 4,–/Disk Ganze Disketten: DM 1,50/Disk Gratisinfo von K. Galz Sonnenbergstr. 55 in 7800 Freiburg

assoziative Datenbank Themadat schnelle, assoziativ arbeitende Datenbank 49,- DM, mit Textmaskengenerator 74,- DM plus 5,- DM Porto (V-Scheck) H. v. Tryller 3200 Hildesheim, Steinbergstr. 6

★ ★ ★ PD-Service DM 6,00 ★ ★ Angebot wie PD dieser Ausgabe Einzeldiskette SS DM 6,00 Pakete 1 – 10, 11 – 20, 21 – 30, 31 – 40 und so w. je DM 38,00 auf je 5 Disketten DS Preis inkl. Disk zzgl. Porto und Verp. DM 5,00 (Ausl. DM 10,00) NN plus 1,70 besser V-Scheck
I K S, Schönblickstr. 7
7500 Karlsruhe 41
ab 18 Uhr 07202/6793

■■ Public-Domain DM 6,00 ■■
■ PD-Angebot dieser Ausgabe ■
■ auf Mark.-Disk SS je DM 6,00 ■
■ + Porto u. Verpack. DM 5,00 ■
■ V.-Scheck o. NN (+DM 1,70) ■
■ J. Rangnow, Talstr. 8 ■
■ 7519 Eppingen (HN) ■
■ 07262/5131 ab 18 Uhr ■
■■ PD-Schnellversand ■■

★ PUBLIC DOMAIN ★
PD aus USA – kein Problem! Demos, Grafik – kein Problem! Spiele, Sound – kein Problem! Sprachen, Sprachausgabe, Anwendungen .... Katalog bestellen bei
Hans Frey
Bruckenäcker 3, 7000 Stuttgart 80

Große Auswahl an: PD-Software Demos professioneller Software An + Verkauf Gebraucht.-Software Info gegen 80 Pf-Marke bei: Ulrike Nolte, Wasenweilerstr. 11a 7817 Ihringen, Tel: 07668/7301

Die beste PD für Ihren ST
★ Jetzt auch doppelseitig ★
EUSAX PD-Service 4018 Langenfeld, Haus Gravener Str. 85

COPY-SCHUTZ IN GFA-BASIC Info gegen 60 Pf bei: ARKOSOFT Postfach 4824 · 4800 Bielefeld 1

DATENVERWALTUNGEN z.B. Vereinsv., Kundenkartei, Lagerv. Video- Schallplatten- Literaturverz., Adressv. ab DM 99,– Luda Software 089/6708355, Staudingerstr. 65, 8 Mü. 83

1. Fußball-Bundesliga-Verw. mit Erg. u. Tab. seit 63 (s/w); HJR Software Evolution, Warendorfer Str. 4, 4722 Ennigerloh 2; 69 DM

★ ★ PD-SOFTWARE ★ ★ Spiele, Musik, Anwendungen, etc. Info bei E. Twardoch, Kaiser-Wilhelm-Str. 88, 1000 Berlin 46

BTX-Manager Atari ST – Btx mit GEM steuern, drucken, Daten in EDV auswerten Drews EDV + Btx *921733# <29> T. 06221-163323

Boris Becker auf PD-Software!! Sowie 50 weitere brandneue Programme gibt's bei Ralf Markert Balbachtal 71, 6970 Lauda

Lehrer: Noten/Korr./Plang. Info: H. Klatt, Pf.340115, 46Dortmund30

+ ST-Software zu Superpreisen + Flight II-129DM, SDI-79DM, Quiwi-64DM, Art Director-159DM, Film Director-179DM, MGT/Mean 18-69DM, V.D.Soft V.Danner/Caspar-Baur Str. 29, 4230 Wesel Sofort kostenloses Info anfordern.
+ + + + + + + + + + + + + + + +

Public-Domain-Blitzversandt! Riesenauswahl! Preise: ab DM 4,– einseitige Disketten ab DM 6,– doppelseitige Inclusive Diskette!!! Auch alle ST-Disks! Auf Doppeldisks beliebig kombinierbar! Gratisliste anfordern bei: A. Gauger Software, Buhlstr. 16a 7505 Ettlingen, 07243/78511

ATARI ST Software in großer Auswahl ab Lager lieferbar Kostenlose Preisliste anfordern HEINICKE-ELECTRONIC, Kommenderiestr. 120, 4500 Osnabrück, Tel. 05 41/8 27 99, Tx 944 966

!!! ACHTUNG !!!
99 Pfennig-Aktion über 250 PD-Disks, Soft- & Hardware Preisknüller .... Gratisinfo bei Ralf Markert, Balbachtalstr. 71, 6970 Lauda, Tel. 09343/8269

SUCHE HARDWARE

ST m. günst. m. viel Zub. 0407127584

SUCHE SOFTWARE

Suche MODULA-2/3.0 M. Class, Saarstr. 10, 7400 Tübingen 6

Software für ST ges. E. Marx Göhren 8834 Pappenheim 09143/951

Suche Prg. od. Tips für Zeitwirtschaft, Plan.-Steuerung z. B. nach Refa-Methodenlehre für Atari ST od. CP/M 2.2 Tel: 06151-716748

TAUSCH

★ ★ ★ Schule + Lernen ★ ★ ★
Lehrer sucht Tauschpartner mögl. Raum Stuttgart. Atari ST und C-64 Werner Knobloch, Gerhardtstr. 19, Tel.: 07031/84188 in 7032 Sindelfingen

KONTAKTE

Suche Kontakte im Raum 7!! M. Pörsch, Karlstr. 34, 727 Nagold

Suche Kontakte + Tauschpartner in Schleswig-Holstein + HH 0451/497267 nach 18 Uhr Thomas

Suche ST-Kontakte T. 0271/352708

Suche dringend Kontakte und Tauschpartner. J. Bullmann, Zur Hindenburgschleuse 3 3 Hannover 71, Tel. 0511/588204

★ ■■■ SAARLAND ■■■ ★
★ Suche Kontakte + Tausch- ★
★ Partner (Softw.) T. 06876/556 ★

VERSCHIEDENES

Diskettenzugriffe bis zu 100 % beschleunigen! Durch kinderleichten Einbau unseres Speedchips! Kein Löten! Ausführliche Anleitung! Gratisinfo anfordern: 0421-6589278. M. Meyer / G. H. Rohlfs-Str. 54c 2820 Bremen 70

SCHWEIZ – TIEFSTPREISE auf allen ATARI + AMIGA Computern, inkl. Peripherie + Software!!! by COMPU-TRADE Emmenstr. 16, CH-3415 Hasle-Rüegsau (034) 614593 auch abends

Achtung! Atari-ST-Fans! Endlich könnt Ihr mit Eurem Hobby Geld verdienen. Wie? Info 02672-2277

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★
★ Schriften für NEC P6/7! ★
★ Über 20 verschiedene Fonts ★
★ Künstlerisches Design. ★
★ Info und Probedrucke von: ★
★ CHARIOT Software ★
★ Rosenhofstr. 15 · Hamburg 6 ★
★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

Weltneuheit für ATARI-ST Video-Film auf Computer Software für Erwachsene Fischer, Pf. 1619, 4830 Gütersloh

Super-Vokabeltrainer mit integrierter Karteiverwaltung! Voll unter GEM! Internationale Zeichen! Viele Lernmodi! Top Handbuch! Gratisinfo anfordern: 0421-6589278 M. Meyer / G. H. Rohlfs-Str. 54c 2820 Bremen 70

BERLIN: Public Domain Software ATARI-Hardware, Software, Zubehör jetzt neu im Computershop Behrendt, Fürbringerstr. 26, 1000 Berlin 61, Tel. 030-6917666

Suche Handbuch: ST-Pascal +, Pro-Fortran, Megamax C 0551/24478

ALADIN mit Roms (neu) DM 290 Tel. (0211) 376956

Super PD-Software & Super-Epromservice! Gratisinfo anfordern: 0421/6589278. M. Meyer / G. H. Rohlfs-Str. 54c, 2820 Bremen 70

■ FUSSBALL-BUNDESLIGA
■ für Atari ST
■ Alle Ergebnisse und denkbaren
■ Tabellen ab 1964. 40,– DM.
■ Gratisinfo anfordern bei
■ Andreas Smoor, Tannenstr. 50
■ 4460 Nordhorn.

>> ST: „Aus 1 mach 2" << Monitor-Umschalteinheit nur 68 DM Kiesenberg Verlag, Pf 579, D-4600 Dortmund 1, Tel. 02 31/57 74 62

Was lange währt, wird endlich...

MCC ST Basic
Version 1.6
Copyright (c) 1986

Metacomco plc.

All Rights Reserved

OK

Endlich ist es da, das neue ST BASIC. Nach schier endloser Wartezeit gibt Atari neuerdings eine von der englischen Firma Metacomco überarbeitete Version des bisher doch sehr dürftigen ST BASIC heraus. Was sich geändert hat und worauf man achten muß, wollen wir Ihnen kurz vorstellen.

Prinzipiell sind beide Versionen des ST BASIC kompatibel zueinander. Man muß jedoch auf einige Dinge Rücksicht nehmen, denn natürlich bestehen Unterschiede zum alten BASIC von Digital Research. Die alten Fehler des DR-BASIC sind, soweit wir das überprüfen konnten, behoben. Es ist sogar etwas schneller geworden, kommt aber an Interpreter der Firmen GFA und Omikron in Sachen Geschwindigkeit, Befehlsumfang und Bedienerführung bei weitem nicht heran. Es gibt eine Vielzahl neuer Befehle, auch bei der Übernahme von alten Programmen muß auf einiges geachtet werden. In Kurzform sieht das so aus:

Der Befehl DEF SEG wurde ganz gestrichen. Er diente zur Bestimmung der Länge von POKE- und PEEK-Befehlen. Stattdessen gibt es nun die Befehle POKE_B, POKE_W, PO-KE_L, PEEK_B, PEEK_W und PEEK_L, die entsprechend ihrer Endung entweder ein Byte, Wort oder Langwort lesen oder schreiben. PEEK und POKE benutzen Integerzahlen, die jetzt automatisch als Longinteger (32 Bit) definiert sind und somit einen Zahlenbereich zwischen -2147483648 und 2147483647 haben.

Bei Zugriffen auf das VDI und AES ist darauf zu achten, daß für VDI wie bisher das Feld CONTRL benutzt wird, dagegen beim AES neue feste Felddefinitionen wie z. B. GEM_ADDRIN Anwendung finden. Eine genaue Übersicht der neuen Worte können Sie in dem Kasten auf der nächsten Seite finden.

Das SYSTAB-Feld hat jetzt Wortlänge (früher Byte), wodurch bei alten Programmen der angegebene Wert halbiert werden muß. Außerdem muß der Wert in Klammern geschrieben werden. So lautet die Anweisung SYSTAB+4 jetzt SYSTAB(2).

Bei der Benutzung der INP-Befehls muß man darauf achten, daß man im Zusammenhang mit -1 nicht immer eine negative Zahl erhält. Das Ergebnis ist aber immer ungleich Null.

Label- und Variablendefinitionen dürfen ab sofort keine Punkte (.) mehr enthalten. Statt dessen kann man aber einen Unterstrich zur besseren Übersicht anwenden.

Das neue alte BASIC bietet dem Anfänger, der nicht gleich tief in die Tasche greifen will, einen billigen Einstieg in die Programmierung des ST. Zu beziehen ist es bei jedem Atari-Fachhändler zum Preis von DM 12,90. Wir bemühen uns, es zum gleichen Preis in unser Softwareangebot aufzunehmen, so daß man es gegebenfalls mit PD-Disketten zusammen bestellen kann. Die Verhandlungen mit Atari sind aber zur Zeit noch nicht abgeschlossen.

(HE)

Die neuen Befehle des MCC BASICS

AREA
- Zeichnet ein ausfüllendes Polygon

ASK MOUSE
- Gibt X,Y-Koordinaten der Maus und Tastenstatus zurück

ASK RGB
- Gibt RGB-Werte zurück

BIOS
- Erlaubt BIOS-Aufrufe

BOX
- Zeichnet eine Box

CLEAR
- Setzt Variablen und Dimensionierungen zurück

DRAW
- Zeichnet eine Linie

DRAWMODE
- Setzt gültigen Zeichenmodus

ERRS
- Enthält Fehlermeldung

GEM_ADDRIN, GEM_ADDROUT, GEM_CONTROL, GEM_GLOBAL, GEM_INTIN, GEM_INTOUT
- vordefinierte AES-Systemvariablen

GEMDOS
- Erlaubt GEMDOS-Aufrufe

GSHAPE
- Malt ein SHAPE-Raster auf den Bildschirm

LINEPAT
- Setzt ein Linienmuster

MAT AREA
- Zeichnet ein gefülltes Polygon aus Datenfeld

MAT DRAW
- Zeichnet eine Linie aus Datenfeld

MAT LINEF
- Zeichnet eine Linie aus Datenfeld

MAT SOUND
- Übergibt Datenfeld an Tongenerator

PATTERN
- Bestimmt ein Füllmuster

PEEK_B, PEEK_W, PEEK_L
- Liest Speicherstelle(n)

POKE_B, POKE_W, POKE_L
- Schreibt in den Speicher

RGB
- Bestimmt RGB-Anteile der Farbpalette

SSHAPE
- Speichert ein SHAPE in ein Feld

STATUS
- Enthält Rückgabeparameter nach Systemaufrufen (TOS, GEM)

XBIOS
- Erlaubt XBIOS-Aufrufe

Inserentenverzeichnis

Vorschau & Impressum

In der nächsten



lesen Sie unter anderem

Das wandelnde Auge
Ein Digitizer im Selbstbau

Nach längerer Entwicklungszeit freuen wir uns Ihnen einen Digitizer präsentieren zu können. Er kann mit einigen interessanten Eigenschaften aufwarten, die wir hier aber noch nicht verraten wollen.

Einer gegen alle, alle gegen einen
Großer Basic-Vergleich

Inzwischen tummeln sich schon viele Basic-Interpreter und auch Compiler auf dem Markt, darunter MCC-, OMIKRON-, GFA-, Fast- und True-Basic. Wir geben einen Überblick und besprechen alle Unterschiede.

Von 8 auf 24 in 4 Sekunden
Wordplus druckt mit 24 Nadeln

Ein großes Manko von Wordplus war bisher, daß man mit 24-Nadel-Druckern keinen brauchbaren Ausdruck erhielt. Unsere Softwarelösung behebt dies auf elegante Weise.

Und es geht doch
ALICE-Pascal – ein Interpreter

Pascal war immer eine Compilersprache. Doch nun ist es geschehen: Ein Pascal-Interpreter ist da. Das bringt natürlich völlig neue Möglichkeiten, die wir näher untersuchten.

Die ST-Computer Ausgabe 10/1987 erscheint am 25.09.87

Impressum

ST-Computer

Verlag:
Heim Fachverlag,
Heidelberger Landstraße 194,
6100 Darmstadt 13,
Telefon (0 61 51) 5 60 57 FAX 0 61 51 / 5 56 89

Verlagsleitung:
Hans-Jörg Heim

Redaktion:
'Merlin' Computer GmbH
Uwe Bartels (UB Chefredakteur)
Marcelo Merino (MM)
Harald Egel (HE)
Harald Schneider (HS)

Anschrift:
'Merlin' Computer GmbH
ST-Computer Redaktion
Postfach 59 69
Industriestr. 26
6236 Eschborn
Tel. 0 61 96 / 48 18 11 FAX 0 61 96 / 4 11 37

Redaktionelle Mitarbeiter:
Markus Nerding (MN), Uli Eickmann (UE), Jürgen Leonhard (JL), Jörg Wilhelm (JW), Oliver Joppich (OJ), Stefan Höhn (SH), Chr. Schormann (CS)

Kommunikation und Presseleitung:
Claus P. Lippert

Autoren dieser Ausgabe:
M. Spehr, C. Lippert, A. Esser, D. Schaun, C. Brod, W. Thomas, Dr. Sarnow

Produktion:
Klaus Schultheis (Ltg.),
Bernd Uwe Failer
Susanne Failer
Bela Kumar
Ilka Lindemann

Anzeigenverkaufsleitung:
Uwe Heim

Anzeigenpreise:
nach Preisliste Nr. 2, gültig ab 1.7.86

Vertrieb:
Hans-Jörg Heim, Uwe Heim,
Heide Schultheis

Erscheinungsweise:
11 x jährlich

Bezugspreis:
Einzelheft DM 6,–.
Jahresabonnement DM 60,– inklusive der gesetzlichen Mehrwertsteuer und den Zustellgebühren für 11 Ausgaben.
Ausland: 80,– DM inkl. Versand

Bezugsmöglichkeiten:
ATARI-Fachhändler, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag unter obiger Adresse.

Druck:
Ferling Druck Darmstadt

Manuskripteinsendungen:
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit ihrer Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern im Heim Verlag. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

Titelseite: Fabian & Mayer
Foto: Rainer Spirandelli